ACL-IJCNLP 2021

The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing

Proceedings of the Conference, Vol. 2 (Short Papers)

August 1 - 6, 2021

Diamond Sponsors

Bloomberg
Engineering

FACEBOOK AI

Google Research

Platinum Sponsors

amazon | science

ByteDance

Megagon Labs

Microsoft

Bai du 百度

DeepMind

Tencent 腾讯

Gold Sponsors

IBM

Alibaba Group
阿里巴巴集团

Silver Sponsors

Bronze Sponsors

©2021 The Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-954085-53-4 (Volume 2)

Message from the General Chair

I am delighted to welcome you to the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)!

We are very grateful for many people. Fei Xia, Wenjie Li (Maggie) and Roberto Navigli, as the Program Chairs, have admirably guided the work of main conference organization and management. The calm and experienced Priscilla Rasmussen has done a lot of work for the signing of contracts with virtual platform company, Underline.io, calculation of registration fees and managing the entire registration process, and communication with sponsors and exhibitors. The amazing 68-person organizing committee, who all contributed so much to make the conference successful: Local Chairs (Priscilla Rasmussen, Thepchai Supnithi, Thanaruk Theeramunkong), Tutorial Chairs (David Chiang, Min Zhang), Workshop Chairs (Kentaro Inui, Michael Strube), Student Research Workshop Chairs (Jad Kabbara, Haitao Lin, Amandalynne Paullada, Jannis Vamvas), Faculty Advisors to the Student Workshop (Jing Jiang, Rico Sennrich, Derek F. Wong, Nianwen Xue), Audio-Video Chairs (Suchathit Boonnag, Rachasak Somyanonthanakul), Conference Handbook Chair (Krit Kosawat), Demonstration Chairs (Heng Ji, Jong C. Park, Rui Xia), Diversity and Inclusion Committee Chairs (Academic Inclusion Chairs: Avirup Sil, Kayathi Chandu, Lifu Huang, Sara Rosenthal; Accessibility Chairs: Minlie Huang, Vivian Chen, Yang Feng; Financial Access Chairs: Martha Yifiru Tachbelie, Alexis Palmer, Ignatius Eziani, Manuel Mager, Nafise Moosavi; Socio-cultural Inclusion Chairs: Alvin Grissom, Xanda Schofield, Pedro Rodriguez), Local Sponsorship Chairs (Rachada Kongkrachantra, Jing Li, Kobkrit Viriyayudhakorn, Zhongyu Wei), Publications Chairs (Yuki Arase, Jing-Shin Chang, Yvette Graham), Publicity Chair (Kai-Fam Wong), Remote Presentation Chairs (Zhongjun He, Nattapol Kritsuthikul, Yadollah Yaghoobzadeh), Sustainability Chairs (Angeliki Lazaridou, Qi Zhang), Reviewer Mentoring Committe Chairs (Jing Huang, Antoine Bosselut, Christophe Gravier), Website and Conference App Chairs (Chutima Beokhaimook, Witchaworn Mankhong), Student Volunteer Coordinator (Dongyan Zhao), Ethic Advisory Committee Chairs (Malvina Nissim, Min-Yen Kan, Xanda Schofield), Social Media Committee Chairs (Luciana Benotti, Lidong Bing, Zhumin Chen, Rachele Sprugnoli, Mark Seligman), Virtual Infrastructure Committee Advisor (Hao Fang), Virtual Infrastructure Committee Chairs (Wei Lu, Krich Nasingkun, Alessandro Raganato, Shaonan Wang, Liang-Chih Yu, Jianfei Yu).

The success of the conference is inseparable from the guidance and advice of ACL Officers. Special thanks to Hinrich Schütze, Rada Mihalcea, David Yarowsky, Shiqi Zhao and Yusuke Miyao. The general chair of NAACL'2021, Dr. Kristina Toutanova provided me much advice based on her experience with NAACL'2021 organization. The friendly cooperation with NAACL'2021 and EACL'2021 workshop chairs and tutorial chairs is very important and is of mutual benefit to each other.

Sponsors and exhibitors are always very important. We are extremely grateful to all sponsors for their continuing support to help our conferences be very successful.

And finally, I would like to thank every one of you for making ACL-IJCNLP'2021 such a success by submitting papers and demos, serving as area chairs and reviewers, session chairs, invited speakers and volunteers, and by joining us in virtual environment.

Welcome and hope you all enjoy the conference!

Chengqing Zong

ACL-IJCNLP'2021 General Chair
June 28, 2021

Message from the Program Chairs

Welcome to the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)! ACL-IJCNLP 2021 has a special historical significance as this is a particularly exciting period: our field has grown dramatically, NLP research is now ubiquitous in products, and the barrier to entry to the field has lowered considerably. Like ACL 2020, ACL-IJCNLP 2021 is held as a virtual conference again due to the worldwide COVID-19 pandemic which has lasted for more than one year. We are very grateful for all of your support and contributions during this difficult time, which make this conference special and memorable.

Abstract and Full-paper Submissions: To synchronize with NAACL 2021, our conference's review cycle was about three weeks shorter than that of ACL 2020. To make the short review cycle work, we introduced an abstract submission step, which required authors to submit an abstract by Jan 25, 2021, one week before the full-paper submission deadline on Feb 1, 2021. This extra step gave NAACL 2021 authors an opportunity to withdraw their papers from NAACL 2021 and submit them to ACL-IJCNLP 2021 based on feedback from NAACL 2021's rebuttal period. In total, we received $4,266$ abstract submissions and $3,350$ full paper submissions.

Tracks: The submissions were assigned to one of 24 topic tracks. The tracks were similar to those used in previous conferences but with a few changes:

1. Based on the number of submissions in previous conferences, we followed NAACL 2021 and combined two tracks ("Semantics: Sentence Level" and "Semantics: Textual Inference and Other Areas of Semantics") into a single track "Semantics: Sentence-level Semantics, Textual Inference and Other areas".

2. To accommodate a wider and more diverse area, we changed the name of the "Computational Social Science and Social Media" track to "Computational Social Science and Cultural Analytics".

3. Following NAACL 2021, we combined the "Theory and Formalism" with the "Cognitive Modeling and Psycholinguistics" areas into "Linguistic theories, Cognitive Modeling and Psycholinguistics". This track is designed to encourage submissions targeted to theoretical underpinning of NLP models which had little/small presence in the past ACL conferences.

4. We introduced a new theme: "NLP for Social Good (NLP4SG)". The application of AI to provide positive social impact has been an important topic in recent years. However, to date, this has not been a topic highlighted at the ACL main conference. This track is designed to invite submissions that can provide insights for the ACL-IJCNLP community on the topic of NLP for Social Good as well as how NLP could potentially cause or be used for social harm.

Program Committee: To meet the reviewer demands of a growing conference without compromising review quality, we started recruiting Senior Area Chairs (SACs) and Area Chairs in early fall 2020. Then we initiated a large-scale reviewer recruiting effort in Nov 2020. We compiled a big list of reviewers from previous conferences, and sent out invitations to more than $9,000$ candidates, asking the ones who were willing to serve to fill out a Microsoft reviewer form. About $4,400$ of the invitees filled out the form. We then worked with SACs and ACs in selecting reviewers and assigning them to appropriate tracks. The whole process of forming the program committee was very complex and took several months to complete and, at the end, we have the largest ever program committee in the history of ACL with 60 SACs, 323 ACs, and $3,685$ primary reviewers.

Reviewer Mentoring Program: Review quality is crucial for the success of a large conference like ACL. Thus, it is of central importance for our community to mentor and train new reviewers in order to keep up with the community's rapid growth, both in terms of submissions and in terms of new members of the community. Therefore, this year we continued the reviewer mentoring program launched with ACL 2020. Ultimately, the goal of this program is to provide long-needed mentoring to new reviewers. We formed a reviewer mentoring committee. Collaborating with them and SACs, we paired Area Chairs (mentors) with first-time ACL reviewers (mentees, often Ph.D. students or junior researchers) during the paper assignment process. The mentees would submit reviews early for the mentors to provide feedback, and the mentees would then revise their reviews based on the feedback. In addition, to help all the reviewers, the reviewer mentoring committee created several videos including the presentation of the mentoring program, a general reviewing tutorial, information about the review form used for this conference, and guidelines on how to consider ethical issues reproducibility in submissions.

Ethical review: The ethical impact and potential applications of our research should be an important consideration for research design, and as artificial intelligence is becoming more mainstream, these issues are increasingly pertinent. To address the potential ethical concerns, we allowed authors to include a broader impact statement or other discussion of ethics in the paper, which does not count towards the page limit. We formed an Ethics Advisory Committee (EAC) with three co-chairs and 57 EAC reviewers. During the review process, reviewers were asked to flag submissions with ethical concerns. The EAC then reviewed all the flagged papers to determine whether the papers should be (a) accepted as is, (b) conditional accepted (with specification of what must be addressed in the camera-ready version in order for the condition to be removed), or (c) rejected on ethical grounds (with explanation of the reject decision). Based on their decisions and the SAC recommendations, we made the accept/reject decisions and sent out acceptance notifications on May 6, 2021. The whole process was explained in a blog posted to the conference website on May 10, 2021. The camera-ready version of the conditionally accepted papers were checked by the EAC again. The EAC informed us that all these papers had made satisfactory revisions and thus we removed the condition on the papers. The whole process was very complex, and we were grateful for the hard work of the EAC and the authors.

Acceptance to Main Conference: After the review process, out of the $3,350$ full submissions, 710 papers (139 short, 571 long) were accepted into the main conference. With an acceptance rate of 21.2%, ACL-IJCNLP 2021 continues to be a highly competitive conference. Based on the nominations from Senior Area Chairs, we selected 28 papers as candidates for the Best Paper awards. We formed a Best Paper Award Committee, who went over all the candidates and selected one best paper, one best theme paper and six outstanding papers.

Findings: To continue the success of Findings at EMNLP 2020, we decided to introduce Findings papers, which are papers that are not accepted for publication in the main conference, but nonetheless have been assessed by the Program Committee as solid work with sufficient substance, quality and novelty. Out of the $3,350$ full submissions, 493 papers were invited to be included in the Findings. Thirty-six papers declined the offer, leading to 457 papers (118 short and 339 long) to be published in the Findings of ACL: ACL-IJCNLP 2021. To increase the visibility of the Finding papers, the authors of such papers can choose to make a 3-minute video to be included in the virtual conference site. Our workshop chairs also helped to pair Findings papers with ACL-IJCNLP 2021 workshops for the possibility of Finding papers to be presented at those workshops.

TACL and CL papers: Continuing the tradition, ACL-IJCNLP 2021 will also feature 27 papers that were published at Transactions of the Association for Computational Linguistics (TACL) and 5 papers from the journal of Computational Linguistics (CL).

Keynote speakers: Another highlight of our program is three exciting keynote talks, given by Prof. Christopher Potts (Stanford University), Prof. Helen Meng (Chinese University of Hong Kong), and Dr. Alejandrina Cristia (École Normale Supérieure).

ACL-IJCNLP 2021 would not be possible without the support from the community. There are many people we would like to thank for their significant contributions! First, we would like to thank our Program Committee, whose names are included in the Program Committee pages in the proceedings:

- Our awesome 60 **Senior Area Chairs** who were instrumental in every aspect of the review process (e.g., AC/reviewer selection, paper assignment, recommendation for paper acceptance, nomination of best papers and outstanding reviewers). For many of them, the scope of their responsibilities was equivalent to chairing a small conference. The 323 **Area Chairs** who led paper review discussions, wrote meta-reviews, and mentored junior reviewers. In addition, they have helped SACs with reviewer selection, paper assignment, and many other tasks.
- Our 3,685 **primary reviewers** and 262 **secondary reviewers** who provided valuable feedback to the authors. Special thanks to those who stepped in at the last minute to serve as emergency reviewers.

Second, we would like to thank many ACL-IJCNLP 2021 committees that we have worked with, including:

- Our **Best Paper Selection Committee**, Bonnie Webber, Tim Baldwin and Ellen Riloff for selecting best papers and outstanding papers under a very tight schedule.
- Our **Ethics Advisory Committee**, chaired by Min-Yen Kan, Malvina Nissim, and Xanda Schofield, for their hard work to ensure that all the accepted papers have addressed the ethical issues appropriately.
- Our **Reviewer Mentoring Committee**, Jing Huang, Antoine Bosselut and Christophe Gravier, for preparing mentoring materials and providing review support to first-time reviewers.
- Our **Publication Co-Chairs**, Jing-Shin Chang, Yuki Arase, and Yvette Graham, for their tremendous effort in making the proceedings.
- Our **Social Media Committee**, chaired by Luciana Benotti, Lidong Bing, Zhumin Chen, Mark Seligman, and Rachele Sprugnoli, for effectively communicating conference updates and other urgent information on social media platforms.
- The **Workshop Chairs**, Kentaro Inui and Michael Strube, for connecting Findings paper authors with individual workshops for possible presentations.
- The **Website & Conference App Chairs**, Chutima Beokhaimook and Witchaworn Mankhong, for making numerous updates to the conference website.

Third, we would like to thank many people who help us with various software used for the conference:

- Rich Gerber at **SoftConf**, who is always quick to respond to our emails and resolve difficulties we encountered with the START system.
- C. M. Downey at the University of Washington, who helped us to extend and run the external paper assignment system developed by Graham Neubig.
- Caterina Lacerra and Rocco Tripodi at the Sapienza University of Rome, who helped us in the creation of internal spreadsheets and processing scripts.
- The whole **Underline** team (Sol Rosenberg, Fun Lee, Jordan Young, Daniel Luise) who created a virtual site for the conference.

As Program chairs, we were in charge of several dozen tasks and many of them were new to us. We would not be able to complete the tasks without the advice from our colleagues, including:

- Our **General Chair** Chengqing Zong, who has been very supportive throughout the whole process, giving us the flexibility to innovate while providing an invaluable sounding board.
- The **Program Co-Chairs of ACL 2020**, Joyce Chai, Natalie Schluter and Joel Tetreault; the **Program Co-Chairs of EMNLP 2020**, Trevor Cohn, Yulan He and Yang Liu; the **Program Co-Chairs of NAACL 2021**, Anna Rumshisky, Luke Zettlemoyer and Dilek Hakkani-Tur, for generously sharing their experience, documentation, and advice in organizing ACL conferences and for answering our questions, often on short notice.
- **ACL Executive Committee**, especially Rada Mihalcea (the **ACL President**) and Hinrich Schütze (the **ACL Past President**), Shiqi Zhao (**Secretary**), Priscilla Rasmussen (**Business Manager**), Nitin Madnani (**Member-at-large**), to help us sort through various issues.
- **TACL Editors-in-Chief** Ani Nenkova and Brian Roark, **TACL Editorial Assistant** Cindy Robinson, and **CL Editor-in-Chief** Hwee Tou Ng for coordinating TACL and CL presentations at the conference.

We would also like to thank all the authors ($8,757$ in total) who submitted their work to the conference. Although we were only able to accept a small percentage of the submissions, your hard work makes this conference exciting and our community strong.

Last, but not least, we thank our students, interns, postdocs, colleagues, and families for being so understanding and supportive when we were swamped by countless conference deadlines and meetings.

Our deepest gratitude is to all of you. We hope you will enjoy the conference.

Fei Xia, University of Washington
Wenjie Li, The Hong Kong Polytechnic University
Roberto Navigli, Sapienza University of Rome

ACL-IJCNLP 2021 Program Committee Co-Chairs

Organizing Committee

General Chair:

Chengqing Zong, Institute of Automation, Chinese Academy of Sciences

Program Committee Co-Chairs:

Wenjie Li, The Hong Kong Polytechnic University
Roberto Navigli, Sapienza University of Rome
Fei Xia, University of Washington

Local Organization Committee Co-Chairs:

Priscilla Rasmussen, Association for Computational Linguistics (ACL)
Thepchai Supnithi, National Electronics and Computer Technology Center (NECTEC)
Thanaruk Theeramunkong, The Artificial Intelligence Association of Thailand and Sirindhorn International Institute of Technology (SIIT), Thammasat University

Tutorial Chairs:

David Chiang, University of Notre Dame
Min Zhang, Soochow University

Workshop Chairs:

Kentaro Inui, Tohoku University
Michael Strube, GmbH Heidelberg

Student Research Workshop Chairs:

Jad Kabbara, McGill University and the Montreal Institute for Learning Algorithms (MILA)
Haitao Lin, Institute of Automation, Chinese Academy of Sciences
Amandalynne Paullada, University of Washington
Jannis Vamvas, Universität Zürich

Faculty Advisors to the Student Research Workshop:

Jing Jiang, Singapore Management University
Rico Sennrich, University of Edinburgh
Derek F. Wong, University of Macau
Nianwen Xue, Brandeis University

Demo Chairs:

Heng Ji, University of Illinois at Urbana-Champaign
Jong C. Park, Korea Advanced Institute of Science and Technology
Rui Xia, Nanjing University of Science and Technology

Publications Chairs:

Yuki Arase, Osaka University
Jing-Shin Chang, National Chi-Nan University
Yvette Graham, Trinity College Dublin

Publicity Chair:

Kai-Fam Wong, The Chinese University of Hong Kong

Sponsorship Co-Chairs:

Rachada Kongkrachantra, Thammasat University
Jing Li, The Hong Kong Polytechnic University
Kobkrit Viriyayudhakorn, iApp Technology Co., Ltd.
Zhongyu Wei, Fudan University

Diversity & Inclusion (D&I) Chairs:

Sub-Committee of Childcare ++ Accessibility:

Leader: Minlie Huang, Tsinghua University
Member: Vivian Chen, National Taiwan University
Member: Yang Feng, Institute of Computing Technology, Chinese Academy of Sciences

Sub-Committee of Academic Inclusion:

Leader: Avirup Sil, IBM
Member: Kayathi Chandu, Carnegie Mellon University
Member: Lifu Huang, Virginia Tech
Member: Sara Rosenthal, IBM Research AI

Sub-Committee of Financial Access:

Leader: Alexis Palmer, University of Colorado Boulder
Leader: Martha Yifiru Tachbelie, Addis Ababa University
Member: Ignatius Eziani, Lancaster University
Member: Manuel Mager, University of Stuttgart
Member: Nafise Moosavi, TU Darmstadt

Sub-Committee of Socio-cultural Inclusion:

Leader: Alvin Grissom, Haverford College
Member: Pedro Rodriguez, University of Maryland, College Park
Member: Xanda Schofield, Harvey Mudd College

Ethics Advisory Committee (EAC):

Min-Yen Kan, National University of Singapore
Malvina Nissim, University of Groningen
Xanda Schofield, Harvey Mudd College

Sustainability Chairs:

Angeliki Lazaridou, DeepMind
Qi Zhang, Fudan University

Audio-Video Chairs:

Suchathit Boonnag, AIAT
Rachasak Somyanonthanakul, Rangsit University

Remote Presentation Chairs:

Zhongjun He, Baidu Co.
Nattapol Kritsuthikul, NECTEC, NSTDA
Yadollah Yaghoobzadeh, University of Tehran

Virtual Infrastructure Committee (VIC):

Advisor:

Hao Fang, Microsoft Semantic Machines

Co-Chairs:

Wei Lu, Singapore University of Technology and Design
Krich Nasingkun, National Electronics and Computer Technology Center
Alessandro Raganato, University of Helsinki
Shaonan Wang, Institute of Automation, Chinese Academy of Sciences
Jianfei Yu, Nanjing University of Science and Technology
Liang-Chih Yu, Yuan Ze University

Reviewer Mentoring Committee Chairs:

Antoine Bosselut, Stanford University
Christophe Gravier, Universite de Saint-Etienne/Lyon
Jing Huang, JD AI Research

Social Media Committee Co-Chairs:

Luciana Benotti, National University of Cordoba
Lidong Bing, DAMO Academy, Alibaba Group
Zhumin Chen, Shandong University
Mark Seligman, Speechmorphing, Inc.
Rachele Sprugnoli, Università Cattolica del Sacro Cuore

Handbook Chair:

Krit Kosawat, NECTEC, NSTDA

Website & Conference App Chairs:

Chutima Beokhaimook, Rangsit University
Witchaworn Mankhong, NECTEC, NSTDA

Student Volunteer Coordinator:

Dongyan Zhao, Peking University

Technical Support:

C. M. Downey, University of Washington
Caterina Lacerra, Sapienza University of Rome
Rocco Tripodi, University of Bologna
Naoki Okada, Osaka University
Masato Yoshinaka, Osaka University

Program Committee

Program Chairs:

Fei Xia, University of Washington
Wenjie Li, The Hong Kong Polytechnic University
Roberto Navigli, Sapienza University of Rome

Senior Area Chairs and Area Chairs:

(Senior area chairs are in bold.)

Computational Social Science and Cultural Analytics:

David Jurgens, **Paolo Rosso**, **Noah Smith**, Timothy Baldwin, Cristina Bosco, Antoine Doucet, Manuel Montes, Alice Oh, Simone Paolo Ponzetto, Sara Rosenthal, Thamar Solorio, Chenhao Tan, Oren Tsur, Leo Wanner, Diyi Yang

Dialogue and Interactive Systems:

Minlie Huang, **Gina-Anne Levow**, **Jason Williams**, Luciana Benotti, Y-Lan Boureau, Yunbo Cao, Asli Celikyilmaz, Yun-Nung Chen, Heriberto Cuayahuitl, Emily Dinan, Maryam Fazel-Zarandi, Kallirroi Georgila, Alborz Geramifard, Matthew Henderson, Ryuichiro Higashinaka, Kentaro Inui, Casey Kennington, Kazunori Komatani, Sungjin Lee, Rebecca J. Passonneau, Giuseppe Riccardi, Ethan Selfridge, Gabriel Skantze, Ruihua Song, David Traum, Stefan Ultes, Tsung-Hsien Wen, Wei Wu, Rui Yan, Kai Yu, Zhou Yu, Wei-Nan Zhang

Discourse and Pragmatics:

Vera Demberg, **Michael Strube**, Jacob Andreas, Chloé Braud, Sadao Kurohashi, Sharid Loáiciga, Nafise Sadat Moosavi

Ethics in NLP:

Ryan Georgi, **Dirk Hovy**, Kai-Wei Chang, Karën Fort, Alvin Grissom II, Margot Mieskes, Vinodkumar Prabhakaran

Information Extraction:

Yunyao Li, **Hoifung Poon**, **Dan Roth**, Alan Akbik, Christos Christodoulopoulos, Leon Derczynski, Jacob Eisenstein, Luheng He, Parisa Kordjamshidi, Mausam, Stephen Mayhew, Makoto Miwa, Lluís Màrquez, Thien Huu Nguyen, Qiang Ning, Haoruo Peng, Roi Reichart, Xiang Ren, Alan Ritter, Alla Rozovskaya, Kevin Small, Yangqiu Song, Vivek Srikumar, Shashank Srivastava, Elior Sulem, Chen-Tse Tsai, William Yang Wang, Wenpeng Yin

Information Retrieval and Text Mining:

Hang Li, **Gabriella Pasi**, Sophia Ananiadou, Mohand Boughanem, Nicola Ferro, Nazli Goharian, Seung-won Hwang, Jing Jiang, Jian-Yun Nie, Raffaele Perego, Suzan Verberne, Quan Wang, Gerard de Melo

Interpretability and Analysis of Models for NLP:

Anna Rogers, **Sameer Singh**, **Xu Sun**, Afra Alishahi, Jasmijn Bastings, Yonatan Belinkov, Danushka Bollegala, Grzegorz Chrupala, Bhuwan Dhingra, Sebastian Gehrmann, Wei Lu, Marco Tulio Ribeiro, Anders Søgaard, Ian Tenney, Byron Wallace

Language Generation:

Michel Galley, **Michael White**, **Jiajun Zhang**, Anya Belz, Giuseppe Carenini, Nina Dethlefs, Mark Dras, Michael Elhadad, Angela Fan, Mary Ellen Foster, Liang Huang, Shujian Huang, Yangfeng Ji, Ioannis Konstas, Sujian Li, Lili Mou, Myle Ott, Ankur P. Parikh, Owen Rambow, Stephen Roller, Advaith Siddharthan, Jinsong Su, Duyu Tang, Zhiguo Wang, Yizhe Zhang

Language Grounding to Vision, Robotics and Beyond:

Mohit Bansal, **Hannaneh Hajishirzi**, Yoav Artzi, Joyce Chai, Nancy Chen, Desmond Elliott, Chuang Gan, Zhe Gan, Ani Kembhavi, Radu Soricut, Jesse Thomason, Mark Yatskar

Linguistic Theories, Cognitive Modeling and Psycholinguistics:

Roger Levy, **James Pustejovsky**, Alexander Clark, Afsaneh Fazly, Naomi Feldman, Tal Linzen, Kyle Mahowald

Machine Learning for NLP:

Ming-Wei Chang, **Kevin Duh**, **Tie-Yan Liu**, **Sebastian Ruder**, Waleed Ammar, Yuki Arase, Niranjan Balasubramanian, Loïc Barrault, Daniel Beck, Yonatan Bisk, Wray Buntine, Allyson Ettinger, Matthias Gallé, Marjan Ghazvininejad, Mohit Iyyer, Shafiq Joty, Sarvnaz Karimi, Hideto Kazawa, Junyi Jessy Li, Zachary Lipton, Yang Liu, Zhiyuan Liu, Daichi Mochihashi, Naoaki Okazaki, Jong Park, Nanyun Peng, Tao Qin, Sujith Ravi, Mrinmaya Sachan, Natalie Schluter, Pontus Stenetorp, Karl Stratos, Jun Suzuki, Lu Wang, Dani Yogatama, Koichiro Yoshino

Machine Translation and Multilinguality:

Philipp Koehn, **Qun Liu**, **François Yvon**, Wilker Aziz, Marine Carpuat, Boxing Chen, Colin Cherry, Marta R. Costa-jussà, Marcello Federico, Yang Feng, Andrew Finch, Mark Fishel, Jiatao Gu, Gholamreza Haffari, Zhongjun He, Mu Li, Liangyou Li, Junhui Li, Kenton Murray, Jan Niehues, Maja Popović, Artem Sokolov, Sara Stymne, Longyue Wang, Tong Xiao

Multidisciplinary and Area Chair COI:

Iryna Gurevych, **Andreas Vlachos**, Dan Goldwasser, Omer Levy, Diarmuid Ó Séaghdha

NLP Applications:

Jimmy Lin, **Vincent Ng**, **Min Zhang**, Beata Beigman Klebanov, Luigi Di Caro, Sanda Harabagiu, Mamoru Komachi, Juntao Li, Jing Li, Yang Liu, David Mimno, Preslav Nakov, Tristan Naumann, Emily Prud'hommeaux, David Smith, Lijun Wu, Jingjing Xu, Min Yang, Jing Yuan, Marcos Zampieri, Wei Zhang

Phonology, Morphology and Word Segmentation:

Yan Song, **Nianwen Xue**, Ryan Cotterell, Xipeng Qiu, Attapol Rutherford

Question Answering:

Jennifer Chu-Carroll, **Alessandro Moschitti**, **Furu Wei**, Roberto Basili, Jordan Boyd-Graber, Weiwei Cheng, Eunsol Choi, Danilo Croce, Li Dong, Yansong Feng, Simone Filice, Radu Florian, Zornitsa Kozareva, Jing Liu, Ramesh Nallapati, Cicero Nogueira dos Santos, Siddharth Patwardhan, Matthias Petri, Oleg Rokhlenko, Minjoon Seo, Avi Sil, Luca Soldaini, Anh Tuan Luu, Olga Uryupina, Thuy Vu, Fabio Massimo Zanzotto

Resources and Evaluation:

Samuel Bowman, **Nancy Ide**, Johan Bos, Tommaso Caselli, Jesse Dodge, Kyle Gorman, Daniel Khashabi, Jin-Dong Kim, Jonathan K. Kummerfeld, John P. McCrae, Joakim Nivre, Massimo Poesio, Saku Sugawara, Adina Williams

Semantics: Lexical:

Mona Diab, **Mohammad Taher Pilehvar**, Marianna Apidianaki, Eduardo Blanco, Jose Camacho-Collados, Manaal Faruqui, Tommaso Pasini, German Rigau, Vered Shwartz, Veselin Stoyanov, Aline Villavicencio, Ivan Vulić, Yadollah Yaghoobzadeh, Yi Zhang

Semantics: Sentence-level Semantics, Textual Inference and Other areas:

Doug Downey, **Raymond Mooney**, **Xiaodan Zhu**, Iz Beltagy, Jonathan Berant, Chandra Bhagavatula, Chris Callison-Burch, Danqi Chen, Greg Durrett, Katrin Erk, Francis Ferraro, Daniel Gildea, Edward Grefenstette, Robin Jia, Douwe Kiela, Mike Lewis, Quan Liu, Christopher Potts, Rachel Rudinger, Mo Yu

Sentiment Analysis, Stylistic Analysis, and Argument Mining:

Bing Liu, **Rada Mihalcea**, **Saif Mohammad**, Alexandra Balahur, Lidong Bing, Julian Brooke, Anna Feldman, Yulan He, Lun-Wei Ku, John Lawrence, Maria Liakata, Smaranda Muresan, Soujanya Poria, Bing Qin, Serena Villata, Xiaojun Wan

Speech and Multimodality:

Haizhou Li, **Florian Metze**, Julia Hockenmaier, Preethi Jyothi, Herman Kamper, Dorothea Kolossa, Hung-yi Lee, Lei Xie

Summarization:

Mirella Lapata, **Horacio Saggion**, Florian Boudin, Jackie Chi Kit Cheung, Katja Filippova, Peter Liu, Fei Liu, Shashi Narayan, Manabu Okumura, Laura Perez-Beltrachini, Maxime Peyrard, Laura Plaza, Xingxing Zhang

Syntax: Tagging, Chunking and Parsing:

Slav Petrov, **Emily Pitler**, Carlos Gómez-Rodríguez, Daniel Hershcovich, Marco Kuhlmann, Yuji Matsumoto, Reut Tsarfaty, Yannick Versley, Yue Zhang, Miryam de Lhoneux

Theme:

Jinho Choi, **Joel Tetreault**, Tim Althoff, Isabelle Augenstein, Steven Bethard, Courtney Napoles, Brendan O'Connor, Yulia Tsvetkov, Rob Voigt

Best Paper Selection Committee:

Timothy Baldwin, Ellen Riloff, Bonnie Webber

Primary Reviewers:

Asma Ben Abacha, Jade Abbott, Ahmed Abdelali, Muhammad Abdul-Mageed, Anne Abeille, Omri Abend, Ahmed AbuRa'ed, Abdalghani Abujabal, Pablo Accuosto, Manoj Acharya, Judit Ács, Heike Adel, Somak Aditya, Stergos Afantenos, Haithem Afli, Sachin Agarwal, Sanchit Agarwal, Shubham Agarwal, Sumeet Agarwal, Rodrigo Agerri, Karan Aggarwal, Piush Aggarwal, Manex Agirrezabal, Željko Agić, Ameeta Agrawal, Priyanka Agrawal, Sweta Agrawal, Gustavo Aguilar, Roee Aharoni, Wasi Ahmad, Natalie Ahn, Lars Ahrenberg, Aman Ahuja, Chaitanya Ahuja, Mohammad Ailannejadi, Akiko Aizawa, Reina Akama, Mohammad Akbari, Alan Akbik, Ahmet Aker, Farhad Akhbardeh, Md. Shad Akhtar, Syed Sarfaraz Akhtar, Adewale Akinfaderin, Nader Akoury, Arjun Akula, Hend Al-Khalifa, Rami Al-Rfou, Nora Al-Twairesh, Fahad AlGhamdi, Firoj Alam, Mehwish Alam, Chris Alberti, Laura Alonso Alemany, Nikolaos Aletras, Jan Alexandersson, Georgios Alexandridis, Mark Alfano, Raquel G. Alhama, Tariq Alhindi, Hamed Alhoori, Malihe Alikhani, Ilseyar Alimova, Afra Alishahi, Tamer Alkhouli, Emily Allaway, Carl Allen, Khalid Alnajjar, Héctor Martínez Alonso, Miguel A. Alonso, Emily Alsentzer, Milad Alshomary, Christoph Alt, Malik Altakrori, Sophia Althammer, Tim Althoff, Tanel Alumäe, Sandra Aluísio, Fernando Alva-Manchego, David Alvarez-Melis, Rami Aly, Marcelo Amancio, Bharat Ram Ambati, Maxime Amblard, Enrique Amigo, Aida Amini, Massih R Amini, Prithviraj Ammanabrolu, Waleed Ammar, Aixiu An, Bo An, Guozhen An, Jisun An, Ashish Anand, Sophia Ananiadou, Raviteja Anantha, Antonios Anastasopoulos, Mark Anderson, Jacob Andreas, Nicholas Andrews, Anietie Andy, Gabor Angeli, Stefanos Angelidis, Luis Espinosa Anke, Diego Antognini, Jean-Yves Antoine, Kaveri Anuranjana, Xiang Ao, Marianna Apidianaki, Emilia Apostolova, Jun Araki, Rahul Aralikatte, Eiji Aramaki, Yuki Arase, Mozhdeh Ariannezhad, Naveen Arivazhagan, Jacob Arkin, Stéphane Aroca-Ouellette, Kushal Arora, Simran Arora, Leila Arras, Ekaterina Artemova, Mikel Artetxe, Philip Arthur, Yoav Artzi, Kristjan Arumae, Ehsaneddin Asgari, Nabiha Asghar, Elliott Ash, Arian Askari, Zhenisbek Assylbekov, Ramón Fernandez Astudillo, Duygu Ataman, Pepa Atanasova, Awais Athar, Giuseppe Attardi, Isabelle Augenstein, Tal August, Eleftherios Avramidis, Ai Ti Aw, Parul Awasthy, Hosein Azarbonyad, Erfan Sadeqi Azer, Wilker Aziz,

Nastaran Babanejad, Rohit Babbar, Bogdan Babych, Nguyen Bach, Ebrahim Bagheri, Parnia Bahar, Ashutosh Baheti, Fan Bai, He Bai, Yu Bai, Yushi Bai, JinYeong Bak, Collin Baker, Vidhisha Balachandran, Alexandra Balahur, Mithun Balakrishna, Anusha Balakrishnan, Oana Balalau, Niranjan Balasubramanian, Ivana Balažević, Ioana Baldini, Timothy Baldwin, Kalika Bali, Miguel Ballesteros, Ramy Baly, Juan Banda, Sivaji Bandyopadhyay, Siddhartha Banerjee, Jeesoo Bang, Seojin Bang, Hritik Bansal, Mohit Bansal, Sameer Bansal, Trapit Bansal, Forrest Sheng Bao, Junwei Bao, Siqi Bao, Yu Bao, Ankur Bapna, Roy Bar-Haim, Mohamad Hardyman Barawi, Edoardo Barba, Adrien Barbaresi, Samuel Barham, Ken Barker, Gianni Barlacchi, Jeremy Barnes, Antonio Valerio Miceli Barone, Loïc Barrault, Valentin Barriere, Alberto Barrón-Cedeño, Max Bartolo, Marco Basaldella, Pierpaolo Basile, Roberto Basili, Ali Basirat, Jasmijn Bastings, Jordi Atserias Batalla, Lisa Bauer, Timo Baumann, William Baumgartner, Susana Bautista, Rachel Bawden, Kathy Baxter, Ian Beaver, Frederic Bechet, Daniel Beck, Lee Becker, Steven Bedrick, Dorothee Beermann, Lisa Beinborn, Ahmad Beirami, Giannis Bekoulis, Núria Bel, Yonatan Belinkov, Eric Bell, Jerome Bellegarda, Meriem Beloucif, Iz Beltagy, Anya Belz, Eyal Ben-David, Luca Benedetto, Luciana Benotti, Adrian Benton, Jonathan Berant, Alexandre Berard, Klaus Berberich, Gábor Berend, Leon

Bergen, Maria Berger, Sabine Bergler, Toms Bergmanis, Rafael Berlanga, Delphine Bernhard, Dario Bertero, Robert Berwick, Laurent Besacier, Steven Bethard, Michele Bevilacqua, Rahul Bhagat, Chandra Bhagavatula, Rasika Bhalerao, Rishabh Bhardwaj, Aditya Bhargava, Archna Bhatia, Parminder Bhatia, Sumit Bhatia, Gantavya Bhatt, Suvrat Bhooshan, Rajarshi Bhowmik, Bin Bi, Wei Bi, Federico Bianchi, Przemyslaw Biecek, Ann Bies, Laura Biester, Yi Bin, Lidong Bing, Alexandra Birch, Steven Bird, Arianna Bisazza, Yonatan Bisk, Johannes Bjerva, Henrik Björklund, Philippe Blache, Eduardo Blanco, Nate Blaylock, Terra Blevins, Rexhina Blloshmi, Su Lin Blodgett, Jelke Bloem, Michael Bloodgood, Théodore Bluche, Valts Blukis, Victoria Bobicev, Praveen Kumar Bodigutla, Ben Bogin, Danushka Bollegala, Valeriia Bolotova-Baranova, Rishi Bommasani, Daniele Bonadiman, Claire Bonial, Francesca Bonin, Ludovico Boratto, Georgeta Bordea, Claudia Borg, Johan Bos, Antal van den Bosch, Cristina Bosco, Antoine Bosselut, Robert Bossy, Nadjet Bouayad-Agha, Florian Boudin, Mohand Boughanem, Gosse Bouma, Zied Bouraoui, Y-Lan Boureau, Samuel R. Bowman, Jordan Boyd-Graber, Johan Boye, Faeze Brahman, António Branco, Jamie Brandon, Kianté Brantley, Pavel Braslavski, Chloé Braud, Felipe Bravo-Marquez, Arthur Bražinskas, Jonathan Brennan, Chris Brew, Thomas Brochhagen, Chris Brockett, Julian Brooke, Samuel Broscheit, Thomas Brovelli (Meyer), Caroline Brun, Dominique Brunato, Luna De Bruyne, Tomáš Brychcín, Yi Bu, Paweł Budzianowski, Sven Buechel, Alberto Bugarín-Diz, Michael Bugert, Trung Bui, Paul Buitelaar, Harry Bunt, Wray Buntine, Greg Burnham, Jill Burstein, Hendrik Buschmeier, Jan Buys, Joan Byamugisha, Bill Byrne, Benjamin Börschinger,

Marco Antonio Sobrevilla Cabezudo, Elena Cabrio, Avi Caciularu, Samuel Cahyawijaya, Deng Cai, Han Cai, Hengyi Cai, Jon Z. Cai, Yi Cai, Andrew Caines, Ruken Cakici, Agostina Calabrese, Iacer Calixto, Chris Callison-Burch, Jesus Calvillo, Jose Camacho-Collados, Erik Cambria, Oana-Maria Camburu, Giovanni Campagna, Leonardo Campillos-Llanos, Niccolò Campolungo, Jon Ander Campos, Ricardo Campos, Burcu Can, Marie Candito, Erion Çano, Guihong Cao, Jiannong Cao, Qingqing Cao, Qingxing Cao, Yanan Cao, Yixin Cao, Yu Cao, Yuan Cao, Yunbo Cao, Ziqiang Cao, Annalina Caputo, Cornelia Caragea, Doina Caragea, Dallas Card, Giuseppe Carenini, Vicente Ivan Sanchez Carmona, Luigi Di Caro, Marine Carpuat, Lucien Carroll, Paula Carvalho, Francisco Casacuberta, Iñigo Casanueva, Helena Caseli, Tommaso Caselli, Vittorio Castelli, Giuseppe Castellucci, Richard Eckart de Castilho, Sheila Castilho, Chundra Cathcart, Andrew Cattle, Paulo Cavalin, Asli Celikyilmaz, Alessandra Cervone, Suchet Chachra, Haixia Chai, Joyce Chai, Abhisek Chakrabarty, Tuhin Chakrabarty, Aishik Chakraborty, Tanmoy Chakraborty, Bharathi Raja Chakravarthi, Gaël de Chalendar, Yllias Chali, Ilias Chalkidis, Nathanael Chambers, Alvin Chan, Hou Pong Chan, Zhangming Chan, Senthil Chandramohan, Muthu Kumar Chandrasekaran, Tai Chang-You, Angel Chang, Baobao Chang, Ernie Chang, Haw-Shiuan Chang, Jing-Shin Chang, Kai-Wei Chang, Ming-Wei Chang, Serina Chang, Yu-Yun Chang, Yung-Chun Chang, Soravit Changpinyo, Guan-Lin Chao, Rajen Chatterjee, Akshay Chaturvedi, Iti Chaturvedi, Stergios Chatzikyriakidis, Aditi Chaudhary, Vishrav Chaudhary, Geeticka Chauhan, Kushal Chawla, Emmanuel Chemla, Bo Chen, Boxing Chen, Chacha Chen, Chung-Chi Chen, Danqi Chen, Daoyuan Chen, Guanyi Chen, Hanjie Chen, Hong-You Chen, Hongshen Chen, Hsin-Hsi Chen, Huimin Chen, Jiaao Chen, Jifan Chen, John Chen, Jun Chen, Kehai Chen, Kezhen Chen, Kuan-Yu Chen, Lei Chen, Lei Chen, Lin Chen, Long Chen, Long Chen, Lu Chen, Luoxin Chen, MeiHua Chen, Meng Chen, Mingda Chen, Muhao Chen, Nancy Chen, Penghe Chen, Qi Chen, Qian Chen, Qianglong Chen, Qingcai Chen, Sanxing Chen, Shizhe Chen, Sihao Chen, Tao Chen, Tongfei Chen, Wenhu Chen, Wenqing Chen, Xilun Chen, Xinchi Chen, Xiuyi Chen, Xiuying Chen, Yang Chen, Yen-Chun Chen, Yi-Chen Chen, Yihong Chen, Yu Chen, Yubo Chen, Yue Chen, Yun Chen, Yun-Nung Chen, Zhenfang Chen, Zhi Chen, Zhiyu Chen, Zhuang Chen, Zhumin Chen, Ziliang Chen, Hao Cheng, Liying Cheng, Lu Cheng, Pengxiang Cheng, Pengyu Cheng, Pu-Jen Cheng, Weiwei Cheng, Xingyi Cheng,

Yong Cheng, Yu Cheng, Vijil Chenthamarakshan, Joe Cheri, Colin Cherry, Emmanuele Chersoni, Jackie Chi Kit Cheung, Jonathan Chevelu, Ethan A. Chi, Zewen Chi, Christian Chiarcos, Jen-Tzung Chien, Hai Leong Chieu, Patricia Chiril, Luis Chiruzzo, Jaemin Cho, Sangwoo Cho, Won Ik Cho, Daejin Choi, Eunsol Choi, Jaesik Choi, Jihun Choi, Jinho D. Choi, Seungtaek Choi, Yejin Choi, Shamil Chollampatt, Jaegul Choo, Leshem Choshen, Prafulla Kumar Choubey, Monojit Choudhury, Khalid Choukri, Jishnu Ray Chowdhury, Koel Dutta Chowdhury, Md Faisal Mahbub Chowdhury, Christos Christodoulopoulos, Fenia Christopoulou, Grzegorz Chrupała, Jennifer Chu-Carroll, Chenhui Chu, Christopher Chu, Zewei Chu, Shun-Po Chuang, Aleksandr Chuklin, Hyung Won Chung, Jin-Woo Chung, Tagyoung Chung, Yi-Ling Chung, Kenneth Church, Abu Nowshed Chy, Manuel Ciosici, Alexander Clark, Christopher Clark, Elizabeth Clark, Kevin Clark, Stephen Clark, Aaron Clauset, Vincent Claveau, Orphee De Clercq, Éric de la Clergerie, Ann Clifton, Miruna-Adriana Clinciu, Maximin Coavoux, Oana Cocarascu, Anne Cocos, Arman Cohan, Edo Cohen-Karlik, Daniel Cohen, Kevin Cohen, Philip Cohen, Trevor Cohn, Marcus Collins, Costanza Conforti, Simone Conia, John Conroy, Danish Contractor, Paul Cook, Bonaventura Coppola, Anna Corazza, Francesco Corcoglioniti, Gonçalo Correia, Caio Corro, Luciano Del Corro, Marta R. Costa-jussà, Ryan Cotterell, Andreas van Cranenburgh, Josep Crego, Alina Maria Cristea, Dan Cristea, Alejandrina Cristia, Danilo Croce, Fabien Cromieres, Paul Crook, James Cross, Tim Van de Cruys, Berthold Crysmann, Montse Cuadros, Heriberto Cuayahuitl, Baiyun Cui, Lei Cui, Leyang Cui, Shaobo Cui, Yiming Cui, Aron Culotta, Iria da Cunha, Washington Cunha, Anna Currey, Tonya Custis,

Wisdom d'Almeida, Jennifer D'Souza, Raj Dabre, Deborah Dahl, Daniel Dahlmeier, Falcon Dai, Xiang Dai, Xinyu Dai, Zeyu Dai, Beatrice Daille, Daniel Dakota, Hercules Dalianis, Siddharth Dalmia, Fahim Dalvi, Marco Damonte, Sandipan Dandapat, Ankit Dangi, Dana Dannells, Abhishek Das, Dipanjan Das, Shouman Das, Pradeep Dasigi, Hal Daumé III, Aida Mostafazadeh Davani, Sam Davidson, Brian Davis, Forrest Davis, Joe Davison, Heidar Davoudi, Johannes Daxenberger, Steve DeNeefe, Jay DeYoung, Alok Debnath, Francien Dechesne, Thierry Declerck, Mathieu Dehouck, Herve Dejean, Sebastien Delecraz, Felice Dell'Orletta, Rodolfo Delmonte, Louise Deléger, Vera Demberg, David Demeter, Seniz Demir, Cagatay Demiralp, Dorottya Demszky, Lingjia Deng, Shumin Deng, Yang Deng, Yue Deng, Yuntian Deng, Zhi-Hong Deng, Pascal Denis, Michael Denkowski, Leon Derczynski, Tyler Derr, Shrey Desai, Nina Dethlefs, Tim Dettmers, Daniel Deutsch, Sunipa Dev, Murthy Devarakonda, Chris Develder, Ann Devitt, Joseph P. Dexter, Sameer Dharur, Paramveer Dhillon, Bhuwan Dhingra, Mona Diab, Shizhe Diao, Gaël Dias, Aniket Didolkar, Emily Dinan, Caiwen Ding, Chenchen Ding, Haibo Ding, Kaize Ding, Liang Ding, Ruixue Ding, Shuoyang Ding, Weicong Ding, Xiao Ding, Zixiang Ding, Liviu P. Dinu, Stefanie Dipper, Anne Dirkson, Nemanja Djuric, Dmitriy Dligach, Simon Dobnik, Jesse Dodge, Charles Dognin, Bill Dolan, Elham Dolatabadi, Miguel Domingo, Lucia Donatelli, Li Dong, MeiX-ing Dong, Ruihai Dong, Xin Dong, Xin Dong, Yue Dong, Longxu Dou, Zi-Yi Dou, Antoine Doucet, C. Downey, Doug Downey, A. Seza Doğruöz, Eduard Dragut, Mark Dras, Markus Dreyer, Rotem Dror, Aleksandr Drozd, Chunning Du, Jiaju Du, Jingfei Du, Jinhua Du, Lan Du, Mengnan Du, Pan Du, Wanyu Du, Yupei Du, Junwen Duan, Nan Duan, Xiangyu Duan, Kumar Dubey, Pablo Duboue, Philipp Dufter, Liam Dugan, Kevin Duh, Ambedkar Dukkipati, Jonathan Dunn, Yoann Dupont, Benjamin Van Durme, Esin Durmus, Nadir Durrani, Greg Durrett, Rory Duthie, Ritam Dutt, Pratik Dutta, Ondřej Dušek, Melody Dye, Chris Dyer, William Dyer, Marc Dymetman, Nouha Dziri,

Haihong E, Kurt Eberle, Sebastian Ebert, Javid Ebrahimi, Daniel Edmiston, Sergey Edunov, Aleksandra Edwards, Steffen Eger, Markus Egg, Koji Eguchi, Yo Ehara, Maud Ehrmann, Vladimir Eidelman, Liat Ein-Dor, Jacob Eisenstein, Asif Ekbal, Asif Ekbal, Wassim El-Hajj,

Yanai Elazar, Maha Elbayad, Heba Elfardy, Ahmed Elgohary, Michael Elhadad, Desmond Elliott, Micha Elsner, Ali Emami, Guy Emerson, Messina Enza, Aykut Erdem, Erkut Erdem, Alexander Erdmann, Akiko Eriguchi, Tomaž Erjavec, Katrin Erk, Liana Ermakova, Patrick Ernst, Marieke van Erp, Carla Parra Escartín, Ramy Eskander, Cristina España-Bonet, Diego Esteves, Dominique Estival, Thierry Etchegoyhen, Allyson Ettinger, Barbara Di Eugenio, Kilian Evang, Richard Evans,

Alexander Fabbri, Guglielmo Faggioli, Farzane Fakhrian, Agnieszka Falenska, Tobias Falke, Angela Fan, Chuang Fan, James Fan, Kai Fan, Yixing Fan, Hao Fang, Hui Fang, Licheng Fang, Rui Fang, Wei Fang, Yimai Fang, Farhood Farahnak, M. Amin Farajian, Oladimeji Farri, Mireia Farrús, Manaal Faruqui, Delia Irazú Hernández Farías, Jean-Philippe Fauconnier, Adam Faulkner, Benoit Favre, Maryam Fazel-Zarandi, Afsaneh Fazly, Amir Feder, Marcello Federico, Guy Feigenblat, Anna Feldman, Naomi Feldman, Sergey Feldman, Junlan Feng, Rui Feng, Shi Feng, Song Feng, Yang Feng, Yansong Feng, Zhangyin Feng, Paulo Fernandes, Daniel Fernández-González, Raquel Fernández, Elisa Ferracane, Francis Ferraro, Thiago Castro Ferreira, Olivier Ferret, Nicola Ferro, Elisabetta Fersini, Oluwaseyi Feyisetan, Anjalie Field, Alejandro Figueroa, Elena Filatova, Simone Filice, Katja Filippova, Andrew Finch, Catherine Finegan-Dollak, Orhan Firat, Mauajama Firdaus, Mark Fishel, Margaret Fleck, Lucie Flek, Dan Flickinger, Michael Flor, Radu Florian, Fabian Flöck, Marina Fomicheva, José A. R. Fonollosa, Erick Fonseca, Marco Aurelio Fonseca, Maxwell Forbes, Tommaso Fornaciari, Karën Fort, Paula Fortuna, George Foster, Mary Ellen Foster, Anette Frank, Robert Frank, Stella Frank, Thomas François, Alexander Fraser, Kathleen C. Fraser, Diego Frassinelli, Dayne Freitag, Markus Freitag, Lea Frermann, Daniel Fried, Annemarie Friedrich, Jason Fries, Guohong Fu, Liye Fu, Tsu-Jui Fu, Zhenxin Fu, Zihao Fu, Zuohui Fu, Akinori Fujino, Yoshinari Fujinuma, Atsushi Fujita, Fumiyo Fukumoto, Nancy Fulda, Adam Funk, Richard Futrell, Michael Färber, Hagen Fürstenau,

Matteo Gabburo, Saadia Gabriel, David Gaddy, Marco Gaido, Núria Gala, Andrea Galassi, Boris Galitsky, Michel Galley, Matthias Gallé, Pablo Gamallo, Michael Gamon, Chuang Gan, Leilei Gan, Yujian Gan, Zhe Gan, Kuzman Ganchev, Sudeep Gandhe, Balaji Ganesan, Devi Ganesan, Suryakanth V Gangashetty, Debasis Ganguly, Cuiyun Gao, Ge Gao, Hanning Gao, Jun Gao, Qiaozi Gao, Shen Gao, Tianyu Gao, Wei Gao, Xiang Gao, Yang Gao, Yang Gao, Yifan Gao, Yingbo Gao, Cristina Garbacea, Diego Garcia-Olano, Eva Martínez Garcia, Marcos Garcia, Matt Gardner, Sarthak Garg, Saurabh Garg, Siddhant Garg, Aparna Garimella, Ekaterina Garmash, Dan Garrette, Milica Gasic, Albert Gatt, Lorenzo Gatti, Manas Gaur, Eric Gaussier, Dipesh Gautam, Vasundhara Gautam, Jidong Ge, Tao Ge, Sebastian Gehrmann, Michaela Geierhos, Alexander Gelbukh, Josef van Genabith, Xinwei Geng, Xiubo Geng, Ryan Georgi, Kallirroi Georgila, Alborz Geramifard, Kim Gerdes, Ulrich Germann, Felix Gervits, Mor Geva, Hamidreza Ghader, Raji Ghawi, Sarik Ghazarian, Marjan Ghazvininejad, Mozhdeh Gheini, Nadia Ghobadipasha, Deepanway Ghosal, Debanjan Ghosh, Sayan Ghosh, Shaona Ghosh, Sourav Ghosh, Sucheta Ghosh, Daniela Gifu, Daniel Gildea, C Lee Giles, Salvatore Giorgi, Voula Giouli, Marco Di Giovanni, Adrià de Gispert, Dimitra Gkatzia, George Gkotsis, Goran Glavaš, Martin Gleize, Kristina Gligoric, Pranav Goel, Rahul Goel, Vaibhava Goel, Nazli Goharian, Seraphina Goldfarb-Tarrant, Anna Goldie, Dan Goldwasser, Sharon Goldwater, Sujatha Das Gollapalli, Marcos Goncalves, Lovedeep Gondara, Heng Gong, Jingjing Gong, Linyuan Gong, Ming Gong, Yeyun Gong, Zhengxian Gong, Ana Valeria González, Jeff Good, Michael Wayne Goodman, Rob van der Goot, Karthik Gopalakrishnan, Jonathan Gordon, Philip John Gorinski, Kyle Gorman, Koustava Goswami, Sourabh Gothe, Cyril Goutte, Amit Goyal, Anuj Goyal, Kartik Goyal, Naman Goyal, Pawan Goyal, Tanya Goyal, Natalia Grabar, Jorge Gracia, Mario Graff, Yvette Graham, Christophe Gravier, Edward Grefenstette, Andrej Zukov Gregoric, David Griol, Yulia Grishina, Ralph Grishman,

Alvin Grissom II, Adam Grycner, Stig-Arne Grönroos, Jia-Chen Gu, Jiatao Gu, Jing Gu, Qing Gu, Shuhao Gu, Yue Gu, Jian Guan, Saiping Guan, Yi Guan, Imane Guellil, Lin Gui, Vincent Guigue, Bruno Guillaume, Liane Guillou, Camille Guinaudeau, Kristina Gulordava, Kalpa Gunaratna, Beliz Gunel, Daya Guo, Han Guo, Honglei Guo, Hongyu Guo, Jiang Guo, Junliang Guo, Qipeng Guo, Quan Guo, Ruocheng Guo, Yinpeng Guo, Yinuo Guo, Zhijiang Guo, Abhinav Gupta, Ankit Gupta, Arpit Gupta, Arshit Gupta, Raghav Gupta, Sonal Gupta, Sparsh Gupta, Vivek Gupta, Iryna Gurevych, Suchin Gururangan, Joakim Gustafson, Ximena Gutierrez-Vasques, Francisco Guzmán, Markus Gärtner, Carlos Gómez-Rodríguez, Jana Götze, Tunga Güngör,

Jung-Woo Ha, Le An Ha, Thanh-Le Ha, Ivan Habernal, Hatem Haddad, Kais Haddar, Asmelash Teka Hadgu, Christian Hadiwinoto, Gholamreza Haffari, Michael Hahn, Udo Hahn, Zhen Hai, Thomas Haider, Jan Hajic, Eva Hajicova, Hannaneh Hajishirzi, Hazem Hajj, Sherzod Hakimov, Kishaloy Halder, Felix Hamborg, William L. Hamilton, Michael Hammond, Thierry Hamon, Jialong Han, Kyu Han, Namgi Han, Ting Han, Wenjuan Han, Xianpei Han, Xiaochuang Han, Xu Han, Abram Handler, Chung-Wei Hang, Viktor Hangya, Tianyong Hao, Rejwanul Haque, Syed Haque, Sanda Harabagiu, Momchil Hardalov, Randy Harris, Mareike Hartmann, Matthias Hartung, Thomas Hartvigsen, Sadid A. Hasan, Peter Hase, Chikara Hashimoto, Saeed-Ul Hassan, Nabil Hathout, Annette Hautli-Janisz, Serhii Havrylov, Hiroaki Hayashi, Katsuhiko Hayashi, Yoshihiko Hayashi, Devamanyu Hazarika, Amir Hazem, Ben He, Hangfeng He, Hao He, Hua He, Jiangen He, Junxian He, Luheng He, Shizhu He, Tianxing He, Xuanli He, Yifan He, Yulan He, Zhengqiu He, Zhongjun He, Kenneth Heafield, Marti A. Hearst, Michael Heck, Behnam Hedayatnia, Johannes Heinecke, Benjamin Heinzerling, Jindřich Helcl, James Henderson, Matthew Henderson, Lisa Anne Hendricks, Simon Hengchen, Leonhard Hennig, Nico Herbig, Christian Herold, Teresa Herrmann, Daniel Hershcovich, Jonathan Herzig, Jack Hessel, Gerhard Heyer, Remu Hida, Christopher Hidey, Djoerd Hiemstra, Ryuichiro Higashinaka, Bertrand Higy, Tsutomu Hirao, Tatsuya Hiraoka, Graeme Hirst, Sorami Hisamoto, Kasia Hitczenko, Lydia-Mai Ho-Dac, Tin Kam Ho, Cong Duy Vu Hoang, Cuong Hoang, Julia Hockenmaier, Johannes Hoffart, Chris Hokamp, Eben Holderness, Nora Hollenstein, Kristy Hollingshead, Laura Hollink, Ari Holtzman, Christopher Homan, Takeshi Homma, Dezhi Hong, Kai Hong, Yu Hong, Mark Hopkins, Enamul Hoque, Helmut Horacek, Ales Horak, Mohammad Javad Hosseini, saghar Hosseini, Veronique Hoste, Feng Hou, Lei Hou, Yufang Hou, Yutai Hou, Dirk Hovy, David M. Howcroft, Christine Howes, Estevam Hruschka, Chao-Chun Hsu, I-Hung Hsu, Wei-Ning Hsu, Phu Mon Htut, Baotian Hu, Bojie Hu, Changjian Hu, Changwei Hu, Chi Hu, Guangneng Hu, Hai Hu, Huang Hu, Jennifer Hu, Jinyi Hu, Mengting Hu, Minghao Hu, Pengwei Hu, Po Hu, Renfen Hu, Wenpeng Hu, Zhe Hu, Zhiting Hu, Ziniu Hu, Xinyu Hua, Yiqing Hua, Chenyang Huang, Chieh-Yang Huang, Chung-Chi Huang, Fei Huang, Guoping Huang, Haoran Huang, Hen-Hsen Huang, Heyan Huang, Jimmy Xiangji Huang, Jing Huang, Jizhou Huang, Kuan-Hao Huang, Liang Huang, Lifu Huang, Luyao Huang, Minlie Huang, Po-Yao Huang, Qingbao Huang, Ruihong Huang, Shujian Huang, Siyu Huang, Xiaolei Huang, Xinting Huang, Xuanjing Huang, Yi-Ting Huang, Yongfeng Huang, Yufang Huang, Zhongqiang Huang, Ziming Huang, Luwen (Vivian) Huangfu, Patrick Huber, Matthias Huck, Kai Hui, Zhen Hui, Ben Hutchinson, Jena D. Hwang, Seung-won Hwang, Sung Ju Hwang, Mika Hämäläinen, Ali Hürriyetoğlu,

Ignacio Iacobacci, Nancy Ide, Adrian Iftene, Oana Ignat, Ryu Iida, Gabriel Ilharco, Filip Ilievski, Dmitry Ilvovsky, Kenji Imamura, Muhammad Imran, Oana Inel, Diana Inkpen, Koji Inoue, Naoya Inoue, Kentaro Inui, Radu Tudor Ionescu, Maxim Ionov, Daphne Ippolito, Tatsuya Ishigaki, Aminul Islam, Tunazzina Islam, Hayate Iso, Dan Iter, Takumi Ito, Lubomir Ivanov, Julia Ive, Tomoya Iwakura, Kenichi Iwatsuki, Srinivasan Iyer, Mohit Iyyer,

Cassandra L. Jacobs, Gilles Jacobs, Jeff Jacobs, Alon Jacovi, Aaron Jaech, Abhyuday Jagannatha, Labiba Jahan, Kokil Jaidka, Prachi Jain, Sarthak Jain, Mimansa Jaiswal, Shoaib Jameel, Abhik Jana, Hyeju Jang, Maciej Janicki, David Janiszek, Sujay Kumar Jauhar, Tommi Jauhiainen, Arun kumar Jayapal, Sébastien Jean, Hwisang Jeon, Sungho Jeon, Minwoo Jeong, Yacine Jernite, Kevin Jesse, Rahul Jha, Donghong Ji, Feng Ji, Yangfeng Ji, Zongcheng Ji, Chen Jia, Robin Jia, Ruipeng Jia, Shengbin Jia, Yuxiang Jia, Zixia Jia, Sittichai Jiampojamarn, Ping Jian, Daxin Jiang, Jing Jiang, Jyun-Yu Jiang, Meng Jiang, Nanjiang Jiang, Zhengbao Jiang, Zhuoren Jiang, Zhuoxuan Jiang, Pengfei Jiao, Wenxiang Jiao, Zhanming Jie, Di Jin, Lifeng Jin, Lisa Jin, Peng Jin, Qin Jin, Xiaolong Jin, Zhijing Jin, Ishan Jindal, Baoyu Jing, Liping Jing, Anna Jobin, Charles Jochim, Anders Johannsen, Richard Johansson, Melvin Johnson, Nebojsa Jojic, Kristiina Jokinen, Erik Jones, Gareth Jones, Siddhartha Reddy Jonnalagadda, Arne Jonsson, Aditya Joshi, Mandar Joshi, Dhanya Jothimani, Shafiq Joty, Meizhi Ju, Xincheng Ju, Yingnan Ju, Jaap Jumelet, Heewoo Jun, Kyomin Jung, Taehee Jung, Zhu Junguo, David Jurgens, Prathyusha Jwalapuram, Preethi Jyothi, Lena Jäger,

Besim Kabashi, Alexandre Kabbach, Jad Kabbara, Sushant Kafle, Sylvain Kahane, Ivana Kajic, Tomoyuki Kajiwara, Mihir Kale, Oren Kalinsky, Aikaterini-Lida Kalouli, Ehsan Kamalloo, Herman Kamper, Jaap Kamps, Min-Yen Kan, Hiroshi Kanayama, Masahiro Kaneko, Jenna Kanerva, Jaewoo Kang, Xiaomian Kang, Katharina Kann, Ryuji Kano, Yoshinobu Kano, Evangelos Kanoulas, Pavan Kapanipathi, Micaela Kaplan, Pinar Karagoz, Alina Karakanta, Svebor Karaman, Giannis Karamanolakis, Siddharth Karamcheti, Mladen Karan, Sarvnaz Karimi, Younes Karimi, Börje Karlsson, Saurav Karmakar, Shubhra Kanti Karmaker, Sanjeev Kumar Karn, Jungo Kasai, Omid Kashefi, Zdeněk Kasner, Nora Kassner, Denys Katerenchuk, Anoop Katti, David Kauchak, Divyansh Kaushik, Pride Kavumba, Daisuke Kawahara, Efsun Sarioglu Kayi, Hideto Kazawa, Ashkan Kazemi, Pei Ke, Katherine Keith, Simon Keizer, Aniruddha Kembhavi, Brendan Kennedy, Casey Kennington, Tom Kenter, Daniel Kershaw, Santosh Kesiraju, Vaibhav Kesri, Madian Khabsa, Shahram Khadivi, Salam Khalifa, Sammy Khalife, Maxim Khalilov, Dinesh Khandelwal, Aparna Khare, Daniel Khashabi, Khalid Al Khatib, Alizishaan Khatri, Chandra Khatri, Tushar Khot, Ashiqur KhudaBukhsh, Douwe Kiela, Halil Kilicoglu, Byeongchang Kim, Donghwan Kim, Gunhee Kim, Hansaem Kim, Hyounghun Kim, Hyunwoo Kim, Jihyuk Kim, Jin-Dong Kim, Joo-Kyung Kim, Jung-Jae Kim, Juyong Kim, Najoung Kim, Seokhwan Kim, Sun Kim, Sundong Kim, Taeuk Kim, Daniel King, Tracy Holloway King, Christo Kirov, Nikita Kitaev, Beata Beigman Klebanov, Ayal Klein, Bennett Kleinberg, Jan-Christoph Klie, Roman Klinger, Julien Kloetzer, Kevin Knight, Alistair Knott, Rebecca Knowles, Miyoung Ko, Hayato Kobayashi, Sosuke Kobayashi, Thomas Kober, Elena Kochkina, Ekaterina Kochmar, Vid Kocijan, Jordan Kodner, Philipp Koehn, Rob Koeling, Svetla Koeva, Mare Koit, Noriyuki Kojima, Dimitrios Kokkinakis, Dorothea Kolossa, Mamoru Komachi, Kazunori Komatani, Rik Koncel-Kedziorski, Grzegorz Kondrak, Fang Kong, Lingkai Kong, Miloslav Konopik, Ioannis Konstas, Parisa Kordjamshidi, Valia Kordoni, Yuta Koreeda, Mandy Korpusik, Katsunori Kotani, Bhushan Kotnis, Fajri Koto, Neema Kotonya, Alexander Kotov, George Kour, Olga Kovaleva, Venelin Kovatchev, Zornitsa Kozareva, Jared Kramer, Bernhard Kratzwald, Sebastian Krause, Elisa Kreiss, Simon Krek, Ralf Krestel, Julia Kreutzer, Amrith Krishna, Kalpesh Krishna, Jayant Krishnamurthy, Rajasekar Krishnamurthy, Nikhil Krishnaswamy, Reno Kriz, Canasai Kruengkrai, Udo Kruschwitz, Anna Kruspe, Germán Kruszewski, Wojciech Kryscinski, Alexander Ku, Lun-Wei Ku, Da Kuang, Marco Kuhlmann, Roland Kuhn, Seth Kulick, Ilia Kulikov, Malhar Kulkarni, Mayank Kulkarni, Artur Kulmizev, Saurabh Kulshreshtha, Abhay Kumar, Abhishek Kumar, Adarsh Kumar, Ashutosh Kumar, Sachin Kumar, Sawan Kumar, Shankar Kumar, Sumeet Kumar, Varun Kumar, Vishwajeet Kumar, Jonathan K. Kummerfeld, Anoop Kunchukuttan, Adhiguna Kuncoro, Souvik Kundu, Florian

Kunneman, Tsung-Ting Kuo, Murathan Kurfalı, Tatsuki Kuribayashi, Mikko Kurimo, Shuhei Kurita, Sadao Kurohashi, Ugur Kursuncu, Aditya Kusupati, Kordula De Kuthy, Mucahid Kutlu, Andrey Kutuzov, Haewoon Kwak, Tom Kwiatkowski, Hongseok Kwon, Arne Köhn,

Caterina Lacerra, Cheng-I Lai, Yuxuan Lai, Chiraag Lala, Divesh Lala, John P. Lalor, Tsz Kin Lam, Wai Lam, Hemank Lamba, Vasileios Lampos, Gerasimos Lampouras, Wuwei Lan, Yunshi Lan, Frédéric Landragin, Phillippe Langlais, Ni Lao, Mirella Lapata, Gabriella Lapesa, Ekaterina Lapshinova-Koltunski, François Lareau, Brian Larson, Stefan Larson, Kornel Laskowski, Mark Last, Luis Lastras, Jey Han Lau, Michael A. Laurenzano, Anne Lauscher, Hady Lauw, Alberto Lavelli, Carolin Lawrence, John Lawrence, Dawn Lawrie, Angeliki Lazaridou, Hung Le, Phong Le, Kevin Leach, Chong Min Lee, Dongkyu Lee, Dongyub Lee, Fei-Tzin Lee, Grandee Lee, Hung-yi Lee, Hwaran Lee, I-Ta Lee, Jay Yoon Lee, Jeong Min Lee, Ji-Ung Lee, Jihwan Lee, Jinhyuk Lee, John Lee, Jongwuk Lee, Kyung-jae Lee, Lung-Hao Lee, Mina Lee, Minwoo Lee, Moontae Lee, Nayeon Lee, Roy Ka-Wei Lee, Sungjin Lee, Yoonhyung Lee, Young-Suk Lee, Els Lefever, Fabrice Lefèvre, Jie Lei, Wenqiang Lei, Jochen L. Leidner, Alessandro Lenci, Yichong Leng, Ben Lengerich, Chee Wee Leong, Yves Lepage, Haley Lepp, Piyawat Lertvittayakumjorn, Gregor Leusch, Jake Lever, Lori Levin, Tomer Levinboim, Rivka Levitan, Sarah Ita Levitan, Gina-Anne Levow, Omer Levy, Ran Levy, Roger Levy, Mike Lewis, Patrick Lewis, Miryam de Lhoneux, Baoli Li, Bei Li, Bryan Li, Chang Li, Chen Li, Cheng-Te Li, Chenliang Li, Dianqi Li, Dongfang Li, Fangtao Li, Fei Li, Feng-Lin Li, Haizhou Li, Hang Li, Hao Li, Haoran Li, Haoran Li, Hongzheng Li, Huayang Li, Irene Li, Jinchao Li, Jing Li, Jiyi Li, Juncheng Li, Junhui Li, Juntao Li, Junyi Jessy Li, Kun Li, Lei Li, Lei Li, Liangyou Li, Manling Li, Maoxi Li, Mu Li, Pan Li, Peifeng Li, Peng Li, Piji Li, Qi Li, Quanzhi Li, Raymond Li, Ruijiang Li, Ruizhe Li, Runnan Li, Shaohua Li, Sheng Li, Shuangyin Li, Si Li, Sujian Li, Tao Li, Tianrui Li, Toby Jia-Jun Li, Wei Li, Wenjie Li, Xiang Li, Xiang Lisa Li, Xiang Lorraine Li, Xiao Li, Xiaoya Li, Xin Li, Xintong Li, Xiujun Li, Xue Li, Yang Li, Yang Li, Yanzeng Li, Yaoyiran Li, Yingjie Li, Yingya Li, Yinqiao Li, Yitong Li, Yuliang Li, Yunyao Li, Zhenghua Li, Zhongyang Li, Zichao Li, Zongxi Li, Maria Liakata, Bin Liang, Chao-Chun Liang, Chen Liang, Davis Liang, Paul Pu Liang, Xiaobo Liang, Xiaodan Liang, Yunlong Liang, Zhicheng Liang, Lizi Liao, Jindřich Libovický, Mohamed Lichouri, Chaya Liebeskind, Luca Di Liello, Constantine Lignos, Anne-Laure Ligozat, Gilbert Lim, Kwan Hui Lim, Nut Limsopatham, Angela Lin, Bill Yuchen Lin, Chenghua Lin, Chin-Yew Lin, Chu-Cheng Lin, Chuan-Jie Lin, Hongfei Lin, Hongyu Lin, Jimmy Lin, Kevin Lin, Kevin Lin, Lucy Lin, Peiqin Lin, Xiang Lin, Yankai Lin, Ying Lin, Zehao Lin, Zhouhan Lin, Zi Lin, Tal Linzen, Marco Lippi, Thomas Lippincott, Zachary Lipton, Pierre Lison, Robert Litschko, Marina Litvak, Bin Liu, Bing Liu, Bing Liu, ChangJian Liu, Chi-Liang Liu, Dayiheng Liu, Dexi Liu, Fangyu Liu, Fei Liu, Fei Liu, Feifan Liu, Haochen Liu, Haokun Liu, Haoyan Liu, Jiachang Liu, Jiahua Liu, Jiangming Liu, Jing Liu, Jingzhou Liu, Kang Liu, Lemao Liu, Ling Liu, Linqing Liu, Maofu Liu, Nelson F. Liu, Peng Liu, Pengfei Liu, Pengfei Liu, Peter Liu, Qian Liu, Qian Liu, Qianchu Liu, Quan Liu, Qun Liu, Tianyi Liu, Tianyu Liu, Tie-Yan Liu, Ting Liu, Weijie Liu, Weiyang Liu, Xianggen Liu, Xiao Liu, Xiaodong Liu, Xuebo Liu, Xueqing Liu, Yan Liu, Yang Liu, Yang Liu, Yang Liu, Ye Liu, Ye Liu, Yijia Liu, Yong Liu, Zemin Liu, Zhenghao Liu, Zhengyuan Liu, Zhengzhong Liu, Zhiyuan Liu, Zhiyuan Liu, Zhuang Liu, Zihan Liu, Zitao Liu, Zoey Liu, Nikola Ljubešić, Kyle Lo, Damien Lolive, Guodong Long, Lucelene Lopes, Marcos Lopes, Jaime Lorenzo-Trueba, Annie Louis, Daniel Loureiro, Ismini Lourentzou, Pablo Loyola, Sharid Loáiciga, Jiasen Lu, Jing Lu, Junyu Lu, Qin Lu, Wei Lu, Yanbin Lu, Yao Lu, Yaojie Lu, Yu Lu, Yi Luan, Nurul Lubis, Alexandra Luccioni, Li Lucy, Cheng Luo, Jiebo Luo, Ling Luo, Ping Luo, Renqian Luo, Robin Luo, Ruotian Luo, Wencan Luo, Yuan Luo, Zhunchen Luo, Anh Tuan Luu, Kelvin Luu, Shangwen Lv, Chunchuan Lyu, Samuel Läubli,

Danni Ma, Jianqiang Ma, Lianbo Ma, Martin Ma, Mingbo Ma, Nianzu Ma, Qianli Ma, Qianwen Ma, Shuming Ma, Tengfei Ma, Wei-Yun Ma, Xiaofei Ma, Xinyin Ma, Xuezhe Ma, Yun Ma, Ismail El Maarouf, Sean MacAvaney, Wolfgang Macherey, Aman Madaan, Avinash Madasu, Mounica Maddela, Nitin Madnani, Andrea Madotto, Walid Magdy, Manuel Mager, Pierre Magistry, Måns Magnusson, Diwakar Mahajan, Suchismit Mahapatra, Adyasha Maharana, Debanjan Mahata, Ayush Maheshwari, Kyle Mahowald, Jean Maillard, Bodhisattwa Prasad Majumder, Navonil Majumder, Peter Makarov, Márton Makrai, Prodromos Malakasiotis, Chaitanya Malaviya, Andreas Maletti, Ankur Mali, Igor Malioutov, Itzik Malkiel, Eric Malmi, Christopher Malon, Rob Malouf, Valentin Malykh, Radhika Mamidi, Emma Manning, Irene Manotas, Elman Mansimov, Saab Mansour, Ramesh Manuvinakurike, Emaad Manzoor, Jiaxin Mao, Runze Mao, Wenji Mao, Yuning Mao, Yuren Mao, Zhendong Mao, Vladislav Maraev, Ana Marasović, Piotr Mardziel, Katerina Margatina, Alda Mari, Benjamin Marie, Alex Marin, Vukosi Marivate, David Martinez, Giovanni Da San Martino, Bruno Martins, Pedro Henrique Martins, Eugenio Martínez-Cámara, Marco Maru, Sameen Maruf, Fiammetta Marulli, Claudia Marzi, Aleksandre Maskharashvili, Maraim Masoud, Matthew Matero, Lambert Mathias, Sandeep Mathias, Nitika Mathur, Prashant Mathur, David Martins de Matos, Sérgio Matos, Yuji Matsumoto, Takuya Matsuzaki, Yevgen Matusevych, Evgeny Matusov, Rowan Hall Maudslay, Mausam, Jonathan May, Stephen Mayhew, Joshua Maynez, Karen Mazidi, Sahisnu Mazumder, Alessandro Mazzei, Diana McCarthy, David McClosky, John P. McCrae, Kate McCurdy, Matthew McDermott, David McDonald, Clifton McFate, Jered McInerney, Bridget McInnes, Kathleen McKeown, Michael McTear, Sara Meftah, Yashar Mehdad, Alexander Mehler, Shikib Mehri, Nikhil Mehta, Sachin Mehta, Sneha Mehta, Clara Meister, Dheeraj Mekala, Gerard de Melo, Julia Mendelsohn, Arul Menezes, Telmo Menezes, Fandong Meng, Rui Meng, Tao Meng, Yu Meng, Zhao Meng, Xue Mengge, Rakesh Radhakrishnan Menon, Amil Merchant, Danny Merkx, Paola Merlo, William Merrill, Mohsen Mesgar, Angeliki Metallinou, Florian Metze, Donald Metzler, Marie-Jean Meurs, Lars Meyer, Adam Meyers, Haitao Mi, Yishu Miao, Yisong Miao, Julian Michael, Lesly Miculicich, Sabrina Mielke, Margot Mieskes, Rada Mihalcea, Todor Mihaylov, Tsvetomila Mihaylova, Nandana Mihindukulasooriya, Claudiu Mihăilă, Martina Miliani, Evangelos Milios, Simon Mille, Corey Miller, Tristan Miller, Alice Millour, Gregory Mills, Emiel van Miltenburg, Eleni Miltsakaki, Farjana Sultana Mim, David Mimno, Bonan Min, Sewon Min, Koji Mineshima, SeyedAbolghasem Mirroshandel, Paramita Mirza, Abhijit Mishra, Pushkar Mishra, Rohan Mishra, Swaroop Mishra, Abhinav Misra, Jeff Mitchell, Verginica Barbu Mititelu, Jelena Mitrović, Sudip Mittal, Vibhu Mittal, Makoto Miwa, Yusuke Miyao, Daichi Mochihashi, Ashutosh Modi, Sarah Moeller, Hans Moen, Aditya Mogadala, Nikita Moghe, Abdelrahman Mohamed, Saif Mohammad, Mahmoud Mohammadi, Alireza Mohammadshahi, Mrinal Mohit, Tasnim Mohiuddin, Michael Mohler, Diego Molla, Francis Mollica, Monica Monachini, Nicholas Monath, Joel Ruben Antony Moniz, Manuel Montes, Emilio Monti, Johanna Monti, Il-Chul Moon, Seungwhan Moon, Raymond Mooney, Andrew Moore, Nafise Sadat Moosavi, Richard Moot, Steven Moran, Erwan Moreau, Antonio Moreno-Ortiz, Jose G. Moreno, Junichiro Mori, Renato De Mori, Véronique Moriceau, Emmanuel Morin, Makoto Morishita, Hajime Morita, John Morris, David R. Mortensen, Ahmadreza Mosallanezhad, Marius Mosbach, Alessandro Moschitti, Masud Moshtaghi, Larry Moss, Lili Mou, Diego Moussallem, Khalil Mrini, Jesse Mu, Jiaqi Mu, Hamdy Mubarak, Pramod Kaushik Mudrakarta, David Mueller, Matteo Muffo, Aldrian Obaja Muis, Animesh Mukherjee, Phoebe Mulcaire, Matthew Mulholland, Benjamin Muller, Philippe Muller, Varish Mulwad, Koji Murakami, Yugo Murawaki, Jamie Murdoch, Smaranda Muresan, Kenton Murray, Rudra Murthy, Shikhar Murty, Tomáš Musil, Rafael Muñoz-Guillena, Agnieszka Mykowiecka, Sheshera Mysore, Lluís Màrquez, Luisa März, Mark-Christoph Müller, Mathias Müller, Thomas Müller,

Anandhavelu N, Farah Nadeem, Nona Naderi, Ryo Nagata, Ajay Nagesh, Aakanksha Naik, Saeed Najafi, Tetsuji Nakagawa, Satoshi Nakamura, Mikio Nakano, Yukiko Nakano, Preslav Nakov, Ramesh Nallapati, Udhyakumar Nallasamy, Feng Nan, Guoshun Nan, Nikita Nangia, Courtney Napoles, Diane Napolitano, Jason Naradowsky, Shashi Narayan, Franco Maria Nardini, Tahira Naseem, Jamal Abdul Nasir, Sudip Naskar, Alexis Nasr, Tristan Naumann, Borja Navarro-Colorado, Roberto Navigli, Mark-Jan Nederhof, Matteo Negri, Isar Nejadgholi, Preksha Nema, Aida Nematzadeh, Ani Nenkova, Guenter Neumann, Mariana Neves, Hwee Tou Ng, Jun-Ping Ng, Vincent Ng, Minh-Quoc Nghiem, Axel-Cyrille Ngonga Ngomo, Dang Tuan Nguyen, Dat Quoc Nguyen, Dong Nguyen, Huyen Nguyen, Kim Anh Nguyen, Thanh Nguyen, Thanh-Tung Nguyen, Thien Huu Nguyen, Toan Q. Nguyen, Truc-Vien T. Nguyen, Trung Hieu Nguyen, Viet-An Nguyen, Jianmo Ni, Eric Nichols, Garrett Nicolai, Massimo Nicosia, Vlad Niculae, Feng Nie, Jian-Yun Nie, Yixin Nie, Jan Niehues, Christina Niklaus, Giannis Nikolentzos, Nikola I. Nikolov, Vassilina Nikoulina, Qiang Ning, Lasguido Nio, Nobal B. Niraula, Kosuke Nishida, Kyosuke Nishida, Noriki Nishida, Masaaki Nishino, Sergiu Nisioi, Malvina Nissim, Tong Niu, Xing Niu, Zheng-Yu Niu, Timothy Niven, Joakim Nivre, Hiroshi Noji, Tadashi Nomoto, Rik van Noord, Damien Nouvel, Jekaterina Novikova, Debora Nozza, Pierre Nugues, Claire Nédellec, Aurélie Névéol,

Alexander O'Connor, Brendan O'Connor, Tim O'Gorman, Daniel Oberski, Jose Ochoa-Luna, Yusuke Oda, Kemal Oflazer, Maciej Ogrodniczuk, Barlas Oguz, Alice Oh, Yoo Rhee Oh, Tomoko Ohkuma, Kiyonori Ohtake, Naoaki Okazaki, Manabu Okumura, Oleg Okun, Hugo Gonçalo Oliveira, Ethel Ong, Yasumasa Onoe, Juri Opitz, Shereen Oraby, Constantin Orasan, Matan Orbach, John Ortega, Petya Osenova, Robert Östling, Naoki Otani, Myle Ott, Zhijian Ou, Hiroki Ouchi, Nedjma Ousidhoum, Jessica Ouyang, Lilja Øvrelid,

Avinesh P.V.S, Deepak P, Maria Leonor Pacheco, Inkit Padhi, Aishwarya Padmakumar, Gustavo Henrique Paetzold, Patrizia Paggio, Arindam Pal, Santanu Pal, Alexis Palmer, Martha Palmer, Endang Pamungkas, Liangming Pan, Xiaoman Pan, Yi-Cheng Pan, Vivek Pandit, Vinay Pandramish, Liang Pang, Richard Yuanzhe Pang, Ludovica Pannitto, Haris Papageorgiou, Pinelopi Papalampidi, Alexandros Papangelis, Nikos Papasarantopoulos, Nikolaos Pappas, Emerson Paraiso, Bhargavi Paranjape, Georgios Paraskevopoulos, Letitia Parcalabescu, Natalie Parde, Antonio Pareja-Lora, Ankur P. Parikh, Haeju Park, Ji Ho Park, Jong Park, Joonsuk Park, Jungsoo Park, Kunwoo Park, Lucy Park, Seong-Bae Park, Serim Park, Sungjoon Park, Youngja Park, Yannick Parmentier, Patrick Paroubek, Ioannis Partalas, Prasanna Parthasarathi, Gabriella Pasi, Tommaso Pasini, Peyman Passban, Rebecca J. Passonneau, Ramakanth Pasunuru, Panupong Pasupat, Raj Patel, Roma Patel, Siddharth Patki, Barun Patra, Braja Gopal Patra, Jasabanta Patro, Viviana Patti, Siddharth Patwardhan, Matthias Paulik, Adam Pauls, Silviu Paun, Ellie Pavlick, John Pavlopoulos, Adam Pease, Pavel Pecina, Ted Pedersen, Jiaxin Pei, Stephan Peitz, Viktor Pekar, Baolin Peng, Hao Peng, Haoruo Peng, Nanyun Peng, Siyao Peng, Wei Peng, Xi Peng, Xutan Peng, Yifan Peng, Gerald Penn, Raffaele Perego, Martin Pereira-Fariña, Lis Kanashiro Pereira, Vittorio Perera, Laura Perez-Beltrachini, Olatz Perez-de-Viñaspre, Gabriele Pergola, Denis Peskov, Ben Peters, Matthew Peters, Matthias Petri, Fabio Petroni, Slav Petrov, Miriam R L Petruck, Maxime Peyrard, Jonas Pfeiffer, Quang Nhat Minh Pham, Maciej Piasecki, Giulio Ermanno Pibiri, Massimo Piccardi, Karl Pichotta, Mohammad Taher Pilehvar, Ildikó Pilán, Tiago Pimentel, Mārcis Pinnis, Juan Pino, Yuval Pinter, Irina Piontkovskaya, Dhivya Piraviperumal, Telmo Pires, Flammie Pirinen, Vito Pirrelli, Miruna Pislar, Emily Pitler, Lidia Pivovarova, Benjamin Piwowarski, Barbara Plank, Lonneke van der Plas, Laura Plaza, Bryan Plummer, Brian Plüss, Lahari Poddar, Nikolaus Poechhacker, Massimo Poesio, Thierry Poibeau, Adam Poliak, Senja Pollak, Lucie Poláková, Girishkumar Ponkiya, Maria Pontiki, Simone Paolo Ponzetto, Hoifung Poon, Kashyap Popat, Maja Popović, Fred Popowich, Soujanya Poria, François

Portet, Christopher Potts, Nima Pourdamghani, Sandhya Prabhakaran, Vinodkumar Prabhakaran, Sameer Pradhan, Animesh Prasad, Judita Preiss, Daniel Preotiuc-Pietro, Ofir Press, Emily Prud'hommeaux, Danish Pruthi, Piotr Przybyła, Michal Ptaszynski, Ratish Puduppully, Rajkumar Pujari, Hemant Purohit, Matthew Purver, James Pustejovsky, Valentina Pyatkin, Juan Antonio Pérez-Ortiz,

Ashequl Qadir, Fanchao Qi, Jianzhong Qi, Dong Qian, Tieyun Qian, Yujie Qian, Chao Qiao, Bing Qin, Guanghui Qin, Lianhui Qin, Libo Qin, Qi Qin, Tao Qin, Liang Qiu, Likun Qiu, Long Qiu, Minghui Qiu, Xipeng Qiu, Yunqi Qiu, Zimeng Qiu, Chen Qu, Yanru Qu, Xiaojun Quan, Martí Quixal,

Ella Rabinovich, Alexandre Rademaker, Gorjan Radevski, Will Radford, Bardia Rafieian, Alessandro Raganato, Preethi Raghavan, Dinesh Raghu, Afshin Rahimi, Zahra Rahimi, Altaf Rahman, Muhammad Rahman, Dheeraj Rajagopal, Shahab Raji, Nitendra Rajput, Taraka Rama, Deepak Ramachandran, Anil Ramakrishna, Ganesh Ramakrishnan, Rohan Ramanath, Owen Rambow, Diego Ramirez-Echavarria, Gabriela Ramirez-de-la-Rosa, Carlos Ramisch, Alan Ramponi, Surangika Ranathunga, Priya Rani, Jinfeng Rao, Yanghui Rao, Ari Rappoport, Ahmad Rashid, Hannah Rashkin, Abhinav Rastogi, Sadaf Abdul Rauf, Vikas Raunak, Shauli Ravfogel, Sujith Ravi, Abhilasha Ravichander, Manikandan Ravikiran, Vinit Ravishankar, Avik Ray, Soumya Ray, Manny Rayner, Paul Rayson, Julia Rayz, Simon Razniewski, Livy Real, Traian Rebedea, Clement Rebuffel, Marta Recasens, Florence Reeder, Ines Rehbein, Georg Rehm, Marek Rei, Roi Reichart, Emily Reif, Paul Reisert, Nils Reiter, Norbert Reithinger, David Reitter, Navid Rekabsaz, Da Ren, Feiliang Ren, Pengjie Ren, Shuhuai Ren, Shuo Ren, Xiang Ren, Yafeng Ren, Yuanhang Ren, Zhaochun Ren, Adithya Renduchintala, Philip Resnik, Luis Reyes-Galindo, Martin Reynaert, Robert Reynolds, Kiamehr Rezaee, Eugénio Ribeiro, Leonardo F. R. Ribeiro, Manuel Sam Ribeiro, Marco Tulio Ribeiro, Corentin Ribeyre, Giuseppe Riccardi, Kyle Richardson, Matthew Richardson, Caitlin Richter, Sebastian Riedel, Martin Riedl, Jason Riesa, German Rigau, Shruti Rijhwani, Matīss Rikters, Laura Rimell, Fabio Rinaldi, Annette Rios, Anthony Rios, Julian Risch, Alan Ritter, Molly Roberts, Gil Rocha, Pedro Rodriguez, Melissa Roemmele, Anna Rogers, Omid Rohanian, Oleg Rokhlenko, Roland Roller, Stephen Roller, Alexey Romanov, Laurent Romary, Salvatore Romeo, Srikanth Ronanki, Wenge Rong, Subendhu Rongali, Francesco Ronzano, Rudolf Rosa, Andrew Rosenberg, Sara Rosenthal, Candace Ross, Sophie Rosset, Paolo Rosso, Aiala Rosá, Dan Roth, Michael Roth, Hossein Rouhizadeh, Masoud Rouhizadeh, Adam Roussel, Joseph Le Roux, Aurko Roy, Subhro Roy, Jos Rozen, Alla Rozovskaya, Raphael Rubino, Sebastian Ruder, Rachel Rudinger, Koustav Rudra, Frank Rudzicz, Jack Rueter, Ivan Vladimir Meza Ruiz, Josef Ruppenhofer, Vasile Rus, Irene Russo, Attapol Rutherford, Tatyana Ruzsics, Max Ryabinin, Maria Ryskina, Hee Jung Ryu, Andreas Rücklé,

Masoud Jalili Sabet, Mrinmaya Sachan, Fatiha Sadat, Arka Sadhu, Mehrnoosh Sadrzadeh, Marzieh Saeidi, Tara Safavi, Sylvie Saget, Horacio Saggion, Benoît Sagot, Koustuv Saha, Monjoy Saha, Punyajoy Saha, Sriparna Saha, Tanay Kumar Saha, Saurav Sahay, Gözde Şahin, Gaurav Sahu, Sunil Kumar Sahu, Keisuke Sakaguchi, Mohammad Salameh, Elizabeth Salesky, Avneesh Saluja, Tanja Samardzic, Rajhans Samdani, Niloofar Safi Samghabadi, Younes Samih, Ramon Sanabria, George Sanchez, Germán Sanchis-Trilles, Victor Sanh, Chinnadhurai Sankar, Sashank Santhanam, Marina Santini, Cicero Nogueira dos Santos, T.Y.S.S Santosh, Bishal Santra, Sebastin Santy, Maarten Sap, Naomi Saphra, Maya Sappelli, Murat Saraclar, Anoop Sarkar, Kamal Sarkar, Prathusha K Sarma, Felix Sasaki, Shota Sasaki, Ryohei Sasano, Danielle Saunders, Agata Savary, Denis Savenkov, Aleksandar Savkov, Ramit Sawhney, Apoorv Saxena, Asad Sayeed, Kevin Scannell, Bianca Scarlini, Carolina Scarton, Thomas Schaaf, Shigehiko Schamoni, Thomas Schatz, Tatjana Scheffler, Yves Scherrer, Timo

Schick, David Schlangen, Dominik Schlechtweg, Viktor Schlegel, Natalie Schluter, Helmut Schmid, Martin Schmitt, Tyler Schnoebelen, Steven Schockaert, Annika Marie Schoene, Mirco Schoenfeld, Alexandra Schofield, Marc Schulder, William Schuler, Claudia Schulz, Hannes Schulz, Elliot Schumacher, Sebastian Schuster, Tal Schuster, Ineke Schuurman, H. Andrew Schwartz, Lane Schwartz, Roy Schwartz, Robert Schwarzenberg, Djamé Seddah, João Sedoc, Abigail See, Elad Segal, Satoshi Sekine, Ethan Selfridge, Thibault Sellam, David Semedo, Olga Seminck, Nasredine Semmar, Cansu Sen, Prithviraj Sen, Shubhashis Sengupta, Rico Sennrich, Minjoon Seo, Yeon Seonwoo, Gwenaelle Cunha Sergio, Abhishek Sethi, Lei Sha, Mahsa Shafaei, Pararth Shah, Samira Shaikh, Igor Shalyminov, Chao Shang, Jingbo Shang, Mingyue Shang, Nan Shao, Yingxia Shao, Yutong Shao, Ori Shapira, Naomi Shapiro, Amr Sharaf, Matthew Shardlow, Abhishek Sharma, Arpit Sharma, Ashish Sharma, Piyush Sharma, Soumya Sharma, Serge Sharoff, Peter Shaw, Lanbo She, Kim Cheng Sheang, Artem Shelmanov, Aili Shen, Dinghan Shen, Gehui Shen, Hua Shen, Jiaming Shen, Qinlan Shen, Sheng Shen, Shiqi Shen, Siqi Shen, Tao Shen, Weizhou Shen, Xiaoyu Shen, Yatian Shen, Yilin Shen, Emily Sheng, Bei Shi, Chuan Shi, Haoyue Shi, Peng Shi, Shuming Shi, Tianze Shi, Weijia Shi, Weiyan Shi, Xiaodong Shi, Xing Shi, Yangyang Shi, Zhan Shi, Zhouxing Shi, Chihiro Shibata, Tomohide Shibata, Anastasia Shimorina, Jamin Shin, Prashant Shiralkar, Boaz Shmueli, Abu Awal Md Shoeb, Linjun Shou, Mohit Shridhar, Manish Shrivastava, Ritvik Shrivastava, Dimitar Shterionov, Kai Shu, Lei Shu, Raphael Shu, Kurt Shuster, Alexander Shvets, Vered Shwartz, Chenglei Si, Mei Si, Aditya Siddhant, Advaith Siddharthan, Georgios Sidiropoulos, Candy Sidner, Melanie Siegel, Avi Sil, Max Silberztein, Max Silberztein, Miikka Silfverberg, Eliezer de Souza da Silva, Fabrizio Silvestri, Michel Simard, Patrick Simianer, Kathleen Siminyu, Goncalo Simoes, Dan Simonson, Matthew Sims, Abhishek Singh, Loitongbam Gyanendro Singh, Sameer Singh, Karan Singla, Priyanka Sinha, Valentina Sintsova, Sunayana Sitaram, Gabriel Skantze, Steve Skiena, Blaž Škrlj, Kevin Small, Koenraad De Smedt, David Smith, Noah A. Smith, Eriks Sneiders, Felipe Soares, Livio Baldini Soares, Artem Sokolov, Luca Soldaini, Aina Garí Soler, Katira Soleymanzadeh, Thamar Solorio, Youngseo Son, Dezhao Song, Haoyu Song, Hyun-Je Song, Kai Song, Kaiqiang Song, Linfeng Song, Ruihua Song, Sanghoun Song, Wei Song, Yan Song, Yangqiu Song, Yiping Song, Rishi Sonthalia, Claudia Soria, Radu Soricut, Aitor Soroa, Alexey Sorokin, Daniil Sorokin, José G. C. de Souza, Marlo Souza, Irena Spasic, Manuela Speranza, Matthias Sperber, Evangelia Spiliopoulou, Andreas Spitz, Rachele Sprugnoli, Mukund Sridhar, Rohini Srihari, Vivek Srikumar, Tejas Srinivasan, Ankit Srivastava, Shashank Srivastava, Edward Stabler, Felix Stahlberg, Sanja Stajner, Ieva Staliūnaitė, Efstathios Stamatatos, Marija Stanojevic, Gabriel Stanovsky, Katherine Stasaski, Shane Steinert-Threlkeld, Georg Stemmer, Pontus Stenetorp, Elias Stengel-Eskin, Evgeny Stepanov, Ian Stewart, Giovanni Stilo, George Stoica, Dario Stojanovski, Kevin Stowe, Veselin Stoyanov, Karl Stratos, Kristina Striegnitz, Michael Strube, Jannik Strötgen, Will Styler, Sara Stymne, Dan Su, Jinsong Su, Keh-Yih Su, Ming-Hsiang Su, Pei-Hao Su, Qinliang Su, Yixuan Su, Yu Su, Nishant Subramani, Aparna Subramanian, Sandeep Subramanian, Sanjay Subramanian, Saku Sugawara, Hiroaki Sugiyama, Alessandro Suglia, Yoshihiko Suhara, Alane Suhr, Dianbo Sui, Zhifang Sui, Octavia-Maria Şulea, Elior Sulem, Md Arafat Sultan, Aixin Sun, Changzhi Sun, Fei Sun, Haitian Sun, Jian Sun, Kai Sun, Le Sun, Ming Sun, Mingming Sun, Si Sun, Simeng Sun, Siqi Sun, Weiwei Sun, Xiaobing Sun, Xu Sun, Yajing Sun, Yawei Sun, Yibo Sun, Yifan Sun, Zequn Sun, Zhiqing Sun, Mujeen Sung, Monica Sunkara, Hanna Suominen, Anshuman Suri, Mirac Suzgun, Hisami Suzuki, Jun Suzuki, Pedro Javier Ortiz Suárez, Sandesh Swamy, Swabha Swayamdipta, Stan Szpakowicz, Ida Szubert, Felipe Sánchez-Martínez, Joan Andreu Sánchez, Diarmuid Ó Séaghdha, Anders Søgaard,

Jeniya Tabassum, Ryuki Tachibana, Marie Tahon, Dima Taji, Ryuichi Takanobu, Sho Takase, David Talbot, Aarne Talman, Ronen Tamari, George Tambouratzis, Aleš Tamchyna, Akihiro

Tamura, Chenhao Tan, Chuanqi Tan, Fei Tan, Jinghua Tan, Jiwei Tan, Liling Tan, Samson Tan, Xu Tan, Buzhou Tang, Duyu Tang, Gongbo Tang, Hao Tang, Jiliang Tang, Jintao Tang, Pingjie Tang, Qingming Tang, Shuai Tang, Siliang Tang, Xiangru Tang, Yi-Kun Tang, Zhiwen Tang, Ludovic Tanguy, Xavier Tannier, Chongyang Tao, Fei Tao, Shiva Taslimipoor, Sandeep Tata, Yuka Tateisi, Rachael Tatman, Michiaki Tatsubori, Marta Tatu, Andon Tchechmedjiev, Christoph Teichmann, Selma Tekir, Serra Sinem Tekiroğlu, Eric Tellez, Ian Tenney, Silvia Terragni, Joel Tetreault, Kapil Thadani, khushboo Thaker, Urmish Thakker, Kilian Theil, Ashok Thillaisundaram, Krishnaprasad Thirunarayan, Jesse Thomason, Brian Thompson, Laure Thompson, Craig Thomson, Camilo Thorne, Yuanhe Tian, Zhiliang Tian, Jörg Tiedemann, Christoph Tillmann, Swati Tiwari, Amalia Todirascu, Takenobu Tokunaga, Gabriele Tolomei, Gaurav Singh Tomar, Nadi Tomeh, Nicholas Tomlin, Marc Tomlinson, Mariya Toneva, Kentaro Torisawa, Marwan Torki, Tiago Timponi Torrent, Juan-Manuel Torres-Moreno, María Inés Torres, Paolo Torroni, Shubham Toshniwal, Samia Touileb, Masashi Toyoda, Amine Trabelsi, Quan Hung Tran, Trang Tran, David Traum, Dietrich Trautmann, Marcos Treviso, Alina Trifan, Rocco Tripodi, Bayu Distiawan Trisedya, Harsh Trivedi, Enrica Troiano, Chen-Tse Tsai, Adam Tsakalidis, Reut Tsarfaty, Bo-Hsiang Tseng, Masaaki Tsuchida, Oren Tsur, Yoshimasa Tsuruoka, Yulia Tsvetkov, Kewei Tu, Lifu Tu, Zhaopeng Tu, Dan Tufis, Iulia Turc, Marco Turchi, Ferhan Ture, Rory Turnbull, Martin Tutek, Elena Tutubalina,

Rutuja Ubale, Ana Sabina Uban, Takuma Udagawa, Stefan Ultes, Bhargav Upadhyay, Zdenka Uresova, Alfonso Ureña-López, Olga Uryupina, Dmitry Ustalov, Masao Utiyama,

Ravi Vadlapudi, Keyon Vafa, Ashwini Vaidya, Vincent Vandeghinste, Keith VanderLinden, Lucy Vanderwende, David Vandyke, Natalia Vanetik, Eva Vanmassenhove, Andrea Vanzo, Shikhar Vashishth, Siddharth Vashishtha, Oleg Vasilyev, Lucy Vasserman, Olga Vechtomova, Luis Gerardo Mojica de la Vega, Julien Velcin, Erik Velldal, Giulia Venturi, Subhashini Venugopalan, Suzan Verberne, Gaurav Verma, Rakesh Verma, Giorgos Vernikos, Yannick Versley, Amir Pouran Ben Veyseh, Marta Vicente, Prashanth Vijayaraghavan, Anvesh Rao Vijjini, David Vilar, David Vilares, Serena Villata, Esau Villatoro-Tello, Aline Villavicencio, Anne Vilnat, Veronika Vincze, Sami Virpioja, Krishnapriya Vishnubhotla, Marco Viviani, Andreas Vlachos, Duy Tin Vo, Ngoc Phuoc An Vo, Tatiana Vodolazova, Nikolai Vogler, Rob Voigt, Soroush Vosoughi, Thuy Vu, Thuy-Trang Vu, Tu Vu, Ivan Vulić, Yogarshi Vyas,

Akifumi Wachi, Henning Wachsmuth, Takashi Wada, Joachim Wagner, Sabine Schulte im Walde, Byron Wallace, Eric Wallace, Mengting Wan, Shengxian Wan, Xiaojun Wan, Yao Wan, Yu Wan, Alex Wang, Bailin Wang, Baoxun Wang, Bin Wang, Bingqing Wang, Boxin Wang, Chang Wang, Changhan Wang, Chao Wang, Cunxiang Wang, Daling Wang, Danqing Wang, Di Wang, Fei Wang, Guangrun Wang, Guoyin Wang, Hai Wang, Han Wang, Han Wang, Hanrui Wang, Hao Wang, Haohan Wang, Haoyu Wang, Heyuan Wang, Hong Wang, Hongfei Wang, Hsin-Min Wang, Hua Wang, Jiaqi Wang, Jin Wang, Jingang Wang, Jingjing Wang, Jingkang Wang, Jingwen Wang, Ke Wang, Kexiang Wang, Liang Wang, Lidan Wang, Longyue Wang, Lu Wang, Lucy Lu Wang, Mengxiang Wang, Mingxuan Wang, Nan Wang, Peifeng Wang, Pidong Wang, Ping Wang, Qiang Wang, Qin Wang, Qingyun Wang, Quan Wang, Rui Wang, Rui Wang, Runze Wang, Shaojun Wang, Shi Wang, Shuai Wang, Shuohang Wang, Tong Wang, Wei Wang, Wei Wang, Wen Wang, Wenbo Wang, Wenhui Wang, Wenqi Wang, Wenxuan Wang, Wenya Wang, William Yang Wang, Xiaozhi Wang, Xin Wang, Xinglong Wang, Xuezhi Wang, Yan Wang, Yaqing Wang, Yequan Wang, Yifei Wang, Yizhong Wang, Yong Wang, Yue Wang, Yujing Wang, Zhen Wang, Zhenyi Wang, Zhichun Wang, Zhiguang Wang, Zhiguo Wang, Zhiqiang Wang, Zhongqing Wang, Zijian Wang, Ziqi Wang, Zirui Wang, Artit Wangperawong, Leo Wanner, Nigel Ward, Alex Warstadt,

Christian Wartena, Zeerak Waseem, Koki Washio, Moshe Wasserblat, Shinji Watanabe, Taro Watanabe, Bonnie Webber, Ingmar Weber, Leon Weber, Noah Weber, Kellie Webster, Jürgen Wedekind, Furu Wei, Jason Wei, Junqiu Wei, Penghui Wei, Wei Wei, Xiaochi Wei, Wang Weiran, Gail Weiss, Charles Welch, Orion Weller, Simon Wells, Haoyang Wen, Lijie Wen, Tsung-Hsien Wen, Peter West, Matthijs Westera, Michael White, Richard Wicentowski, Michael Wiegand, John Wieting, Gijs Wijnholds, Ethan Wilcox, Rodrigo Wilkens, Adina Williams, Jake Williams, Jason D Williams, Jennifer Williams, Steven Wilson, Shuly Wintner, Sam Wiseman, Dawid Wisniewski, Guillaume Wisniewski, Tomer Wolfson, Marcin Woliński, Derek F. Wong, Ka Ho Wong, Tak-Lam Wong, Dina Wonsever, Zach Wood-Doughty, Alina Wróblewska, Bowen Wu, Changxing Wu, Chien-Sheng Wu, Fangzhao Wu, Junshuang Wu, Ledell Wu, Lijun Wu, Lingfei Wu, Shih-Hung Wu, Shijie Wu, Tongshuang Wu, Wei Wu, Xianchao Wu, Xixin Wu, Yen-Chen Wu, Youzheng Wu, Yu Wu, Yuanbin Wu, Yuexin Wu, Yuting Wu, Yuxiang Wu, Zeqiu Wu, Zhanghao Wu, Zhen Wu, Zhiyong Wu, Joern Wuebker, Christian Wurm,

Congying Xia, Fei Xia, Jingbo Xia, Mengzhou Xia, Patrick Xia, Qingrong Xia, Rui Xia, Yingce Xia, Yikun Xian, Chaojun Xiao, Huiru Xiao, Lin Xiao, Tong Xiao, Wen Xiao, Xinyan Xiao, Yanghua Xiao, Boyi Xie, Jun Xie, Lei Xie, Qianqian Xie, Ruobing Xie, Bowen Xing, Chen Xing, Frank Xing, Chao Xiong, Hao Xiong, Hongyu Xiong, Wenhan Xiong, Benfeng Xu, Boyan Xu, Can Xu, Chang Xu, Chen Xu, Chenchen Xu, Frank F. Xu, Guandong Xu, Hongfei Xu, Jiacheng Xu, Jinan Xu, Jingjing Xu, Jun Xu, Lei Xu, Lu Xu, Mingzhou Xu, Peng Xu, Qiongkai Xu, Wei Xu, Weiran Xu, Wenduan Xu, Xinnuo Xu, Yan Xu, Yang Xu, Yumo Xu, Yunqiu Xu, Zenglin Xu, Zhen Xu, Huichao Xue, Nianwen Xue,

Mohit Yadav, Shweta Yadav, Yadollah Yaghoobzadeh, Mohamed Yahya, Ikuya Yamada, Ivan Yamshchikov, Jun Yan, Lingyong Yan, Ming Yan, Rui Yan, Yu Yan, Zhao Yan, Baosong Yang, Bishan Yang, Chenghao Yang, Diyi Yang, Haiqin Yang, Jaewon Yang, Jie Yang, Jun Yang, Junjie Yang, Liner Yang, Linyi Yang, Liu Yang, Min Yang, Muyun Yang, Nan Yang, Qian Yang, Sen Yang, Tsung-Yen Yang, Wei Yang, Weiwei Yang, Wenmian Yang, Yaqin Yang, Yazheng Yang, Yiben Yang, Yilin Yang, Zhichao Yang, Zixiaofan Yang, Ziyi Yang, Tae Yano, He Yanqing, Huaxiu Yao, Jin-Ge Yao, Liang Yao, Wenlin Yao, Yiqun Yao, Mark Yatskar, Semih Yavuz, Deming Ye, Hai Ye, Qinyuan Ye, Xiaoyuan Yi, Wen-wai Yim, Seid Muhie Yimam, Da Yin, Haiyan Yin, Qingyu Yin, Wenpeng Yin, Xuwang Yin, Yichun Yin, Anssi Yli-Jyrä, Michael Yoder, Dani Yogatama, Sho Yokoi, Zheng Xin Yong, Seunghyun Yoon, Masashi Yoshikawa, Naoki Yoshinaga, Koichiro Yoshino, Steve Young, Bei Yu, Bowen Yu, Changlong Yu, Chen Yu, Dian Yu, Dian Yu, Dong Yu, Heng Yu, Hong Yu, Jianfei Yu, Jifan Yu, Juntao Yu, Kai Yu, Licheng Yu, Mo Yu, Ping Yu, Seunghak Yu, Tao Yu, Wenhao Yu, Wenmeng Yu, Xiaodong Yu, Zhou Yu, Caixia Yuan, Jianhua Yuan, Nicholas Jing Yuan, Xingdi Yuan, Zheng Yuan, François Yvon,

Menno van Zaanen, Wajdi Zaghouani, Farooq Zaman, Mohammadzaman Zamani, Marcos Zampieri, Yuan Zang, Fabio Massimo Zanzotto, Alessandra Zarcone, Gian Piero Zarri, Sina Zarrieß, Vicky Zayats, Omnia Zayed, Rabih Zbib, Albin Zehe, Amir Zeldes, Rowan Zellers, Yury Zemlyanskiy, Daojian Zeng, Jiali Zeng, Weixin Zeng, Xiangrong Zeng, Xingshan Zeng, Zhaohao Zeng, Deniz Zeyrek, Hanwen Zha, Sheng Zha, Fangzhou Zhai, Shuang (Sophie) Zhai, Yuming Zhai, Biao Zhang, Boliang Zhang, Bowen Zhang, Bowen Zhang, Chao Zhang, Chenbin Zhang, Chenwei Zhang, Chuheng Zhang, Dong Zhang, Dongxu Zhang, Dongyu Zhang, Hainan Zhang, Hao Zhang, Haoyu Zhang, Hongming Zhang, Huijun Zhang, Jiajun Zhang, Jianguo Zhang, Jinchao Zhang, Jingqing Zhang, Jipeng Zhang, Ke Zhang, Kun Zhang, Kunpeng Zhang, Lei Zhang, Licheng Zhang, Longtu Zhang, Meishan Zhang, Meng Zhang, Michael Zhang, Min Zhang, Ningyu Zhang, Qi Zhang, Richong Zhang, Rui Zhang,

Ruiyi Zhang, Ruqing Zhang, Shaohua Zhang, Sheng Zhang, Shujian Zhang, Shuo Zhang, Tongtao Zhang, Wei Emma Zhang, Wei Zhang, Wei-Nan Zhang, Weiwei Zhang, Wen Zhang, Xiang Zhang, Xiang Zhang, Xiangliang Zhang, Xiao Zhang, Xiaotong Zhang, Xiaoying Zhang, Xingxing Zhang, Xinsong Zhang, Xinyuan Zhang, Xuanwei Zhang, Xuanyu Zhang, Xuchao Zhang, Yi Zhang, Yi Zhang, Yi Zhang, Yichi Zhang, Yifan Zhang, Yizhe Zhang, Yu Zhang, Yuan Zhang, Yuan Zhang, Yue Zhang, Yunyi Zhang, Yuqi Zhang, Yuyu Zhang, Zequn Zhang, Zeyu Zhang, Zhe Zhang, Zheng Zhang, Zhirui Zhang, Zhisong Zhang, Zhuosheng Zhang, Chao Zhao, Chen Zhao, Dongyan Zhao, Fei Zhao, Guangxiang Zhao, Jie Zhao, Jieyu Zhao, Jieyu Zhao, Jun Zhao, Kai Zhao, Lujun Zhao, Mengjie Zhao, Sanqiang Zhao, Tiancheng Zhao, Tianyu Zhao, Tiejun Zhao, Wei Zhao, Yang Zhao, Yanpeng Zhao, Yanyan Zhao, Yao Zhao, Yinggong Zhao, Zhou Zhao, Baigong Zheng, Bo Zheng, Changmeng Zheng, Lin Zheng, Renjie Zheng, Xin Zheng, Yinhe Zheng, Ming Zhong, Peixiang Zhong, Victor Zhong, Zexuan Zhong, Ben Zhou, Chunting Zhou, Dong Zhou, Ganbin Zhou, Giulio Zhou, Guangyou Zhou, Hao Zhou, Jiawei Zhou, Jie Zhou, Jingbo Zhou, Junpei Zhou, Junru Zhou, Junsheng Zhou, Junwei Zhou, Li Zhou, Long Zhou, Mantong Zhou, Pei Zhou, Qiji Zhou, Qingyu Zhou, Shuchang Zhou, Shuyan Zhou, Wangchunshu Zhou, Wenxuan Zhou, Xiang Zhou, Xiangyang Zhou, Xuhui Zhou, Yichao Zhou, Yilun Zhou, Zhengyu Zhou, Zhihan Zhou, Zhong Zhou, Dawei Zhu, Haichao Zhu, Henghui Zhu, Jia Zhu, Jinhua Zhu, Junnan Zhu, Kenny Zhu, Ligeng Zhu, Muhua Zhu, Pengfei Zhu, Su Zhu, Wanzheng Zhu, Wei Zhu, Xiaodan Zhu, Xiaofeng Zhu, Zining Zhu, Fuzhen Zhuang, Honglei Zhuang, Yimeng Zhuang, Yuan Zhuang, Leonardo Zilio, Roger Zimmermann, Heike Zinsmeister, Ayah Zirikly, Imed Zitouni, Ran Zmigrod, Michael Zock, Shi Zong, Markus Zopf, Bowei Zou, Yanyan Zou, Amal Zouaq, Arkaitz Zubiaga, Frederike Zufall.

Secondary Reviewers:

Salah Ait-Mokthar, Eunice Akani, Zainab Albujasim, Nada Aldarrab, Sherlon Almeida, Chantal Amrhein, Nikolay Arefyev, Siddhant Arora,

Pablo Badilla, Jorge Balazs, Hubert Baniecki, Hongchang Bao, Liao Baohao, Loïc Barrault, Anton Belyy, Nathan Berger, Aditya Bhargava, Shaily Bhatt, Nikita Bhutani, Yonatan Bitton, Rexhina Blloshmi, Janos Borst, Fabienne Braune, Max Bryan, Ana-Maria Bucur, Wray Buntine, Kim Bürgl,

Hongjie Cai, Jiangxia Cao, Rémi Cardon, Steffen Castle, Sophia Chan, Piyush Chawla, Siva Uday Sampreeth Chebolu, Fumian Chen, Zitong Cheng, Donghee Choi, Eric Corlett,

Jamell Dacon, Leonard Dahlmann, Yinpei Dai, Dhairya Dalal, Maxime D. Armstrong, Souvik Das, Loic De Langhe, Johannes Deleu, Marco Del Treidici, Lorenzo De Mattei, maureen de seyssel, Anurag Deshmukh, Nina Dethlefs, Hannah Devinney, Juglar Diaz, Bayu Distiawan Trisedya, Suman Dowlagar, Rotem Dror, Andrew Drozov, Nan Duan,

Liana Ermakova,

Marzieh Fadaee, Joachim Fainberg, Nils Feldhus, Katy Felkner, Andrew Finch, Clémentine Fourrier,

Xiubo Geng, Efthymios Georgiou, Iacopo Ghinassi, Behrooz Ghorbani, Christian Gollan, Ming Gong, Alicja Gosiewska, Tamas Grosz, Yu Gu, Shu Guo, Ashim Gupta,

Marius Hamacher, Kijong Han, Bradley Hauer, Hangfeng He, Michael Heck, Felix Helfer, Nils Holzenberger, Weiwei Hou, Weronika Hryniewska, Zechuan Hu, Xinting Huang, Yeh Hui-Syuan, Yongkeun Hwang,

Radu Iacob, Nikolai Ilinykh,

Gilles Jacobs, Aman Jaiswal, Anubhav Jangra, Minbyul Jeong, Ryan J. Hubbard, jianshu Ji, Qi Jia, Hao Jiang, Bernal Jimenez Gutierrez, Arne Jönsson,

Tai-lin Karidi, Hemant Kathania, Divyansh Kaushik, Gangwoo Kim, Guillaume Klein, Mateusz Klimaszewski, Xenia Klinge, Ryosuke Kohita, Michael Kozielski, Akshay Krishna Sheshadri, Shachi H. Kumar, Yaman Kumar, Nicholas Kuo, Kemal Kurniawan, Heeyoung Kwak,

Philippe Laban, Samuel Larkin, Hung-yi Lee, Juho Leinonen, Gael Lejeune, Bai Li, Jinggui Liang, Yaqing Liao, Ruogu Lin, Alisa Liu, Kaiji Lu,

Danni Ma, Avinash Madasu, Arnob Mallik, Ramesh Manuvinakurike, Chengsheng Mao, Mounika Marreddy, Federico Martelli, Taha Masood, Diego Maupomé, Matt McNeill, Laiba Mehnaz, Alessio Miaschi, Alice Millour, Flor Miriam Plaza del Arco, Ishani Mondol, Víctor M. Sánchez-Cartagena, Philipp Müller, Deepak Muralidharan, Toshiki Muromachi,

Kouta Nakayama, Yatin Nandwani, Sara Ng, Dan Nguyen,

Mayumi Ohta, Eda Okur, Siru Ouyang, Nadav Oved, Nanami Ozawa,

Vardaan Pahuja, Margherita Pallottino, Jiaxin Pan, Subhadarshi Panda, Jianhui Pang, Andrea Papaluca, Nivranshu Pasricha, Archita Pathak, Chen (Patrick) Pei, Jiahuan Pei, Qianqian Peng, MinhQuang Pham, Joan Plepi, Luigi Procopio,

Weizhen Qi, Yi Qin,

Dheeraj Rajagopal, Alan Ramponi, Fanny Rancourt, Danial Raza, Evelina Rennes, Matías Rojas, Alexis Ross, Aku Rouhe, Hossein Rouhizadeh, Cao Rui,

Sougata Saha, Naveen Saini, Flora Sakketou, Tanja Samardzic, Brenda Santana, Twisampati Sarkar, Shiki Sato, Shigehiko Schamoni, Lena Schiffer, Elad Segal, Sina Semnani, Sandaru Seneviratne, Hendra Setiawan, Kyle Shaffer, Sanket Shah, Jiawei Sheng, Jiatong Shi, Linjun Shou, Keshav Singh, Gabriella Skitalinskaya, Nikita Soni, Anna Sotnikova, Olga Sozinova, Anirudh Srinivasan, Tomasz Stanislawek, Kevin Stier, Peng Su, Shivashankar Subramanian, Yanming Sun, Shahbaz Syed,

Mohsen Tabasy, Ryo Takasu, Duyu Tang, Marc Tanti, Maksym Taranukhin, Xanh Thi Ho, Evgeniia Tokarchuk, Thanh Tran, Yang Trista Cao, Henry Tsai, An Tuan Dao,

Clara Vania, Benjamin van Niekerk, Suzan Verberne, Huy Vu,

Manya Wadhwa, AbdelRahman Wael, Cheng Wang, Sabine Weber, Cyril Weerasoriya, Andreas Weise, Zhihua Wen, Taesun Whang, Katarzyna Woźnica, Liangqing Wu,

Xiaolin Xia, Yuqing Xie, Benfeng Xu,

Brian Yan, Jenny Yang, Xinzhi Yao, Yongjing Yin, Zheng-Xin Yong, Jaehyo Yoo, Ori Yoran, Bowen Yu, Weizhe Yuan,

Frank D. Zamora-Reina, Najam Zaidi, Klim Zaporojets, Shuxi Zeng, Thomas Zenkel, Runzhe Zhan, Chen Zhang, Jinman Zhao, Houquan Zhou, Zining Zhu, Franziska Zimmermann, Elaine Zosa, Jie Zou, Xinxing Zu.

We would like to recognize the following Outstanding Reviewers:

Rami Al-Rfou, Carl Allen, Mark Anderson, Stefanos Angelidis, Jean-Yves Antoine, Leila Arras,

Rohit Babar, Hritik Bansal, Su Lin Blodgett, Valts Blukis, Nadjet Bouayad-Agha, Arthur Bražinskas, Michael Bugert,

Vittorio Castelli, Hou Pong Chan, Fenia Christopoulou, Elizabeth Clark, Kevin Clark, Vincent Claveau, Anna Currey,

Hal Daume III, Forrest Davis, Steve DeNeefe, Daniel Deutsch, Sunipa Dev, Joseph P. Dexter, Pablo Duboue, Philip Dufter, Ondřej Dušek, Rory Duthie, Nouha Dziri,

Alexander Fabbri, Agnieszka Falenska, Sergey Feldman, Daniel Fernandez-Gonzalez, Anjalie Field, Margret Fleck, Michael Flor, Maxwell Forbes, Thomas Francois, Daniel Fried, Zhenxin Fu,

Matteo Gabburo, Yang Gao, Siddhant Garg, Aina Garí Soler, Marcos Goncalves, Jana Götze, Bruno Guillaume,

Xiaochuang Han, Peter Hase, Hiroaki Hayashi, Devamanyu Hazarika, Jack Hessel, Tsutomu Hirao, Ari Holtzman, Xuanjing Huang,

Gabriel Ilharco,

Gilles Jacobs, Alon Jacovi, Sarthak Jain, Nanjiang Jiang, Anders Johanssen,

Jaap Kamps, Siddharth Karamcheti, Brendan Kennedy, Jihyuk Kim, Byeongchang Kim, Nikita Kitaev, Hayato Kobayashi, Noriyuki Kojima, Seth Kulick, Sawan Kumar, Adhiguna Kuncoro,

Jake Lever, Yaoyiran Li, Jindřich Libovický, Fangyu Liu,

Wei-Yun Ma, Adyasha Maharana, Alexander Mehler, Sabrina J. Mielke, Evangelios Milios, Sewon Min, Jeff Mitchell,

Matan Orbach, Jessica Ouyang,

Aishwarya Padmakumar, Bhargavi Paranjape, Letitia Parcalabescu, Carla Parra Escartín, Viviana Patti, Karl Pichotta, Tiago Pimentel, Lahari Poddar, Rajkumar Pujar,

Xiaojun Quan,

Shuhuai Ren, Philip Resnik, Gil Rocha,

Sylvie Saget, Victor Sanh, Timo Schick, Tyler Schnoebelen, Roy Schwartz, Abigail See, Rico Sennrich, Peter Shaw, Qinlan Shen, Tianze Shi, Valentina Sintsova, Wei Song, Youngseo Song, Andreas Spitz, Yoshihiko Suhara, Alane Suhr,

Ronen Tamari, Yuanhe Tian,

Rob van der Goot, Emiel van Miltenberg, Rik van Noord, Lucy Vanderwende, David Vilares,

Alex Wang, Zijian Wang, Zhen Wang, Alex Warstadt, Gail Weiss, Alina Wróblewska, Jorn Wuebker,

Jiacheng Xu,

Michael Yoder, Naoki Yoshinaga, Steve Young, Dian Yu,

Wei Zhang, Zeyu Zhang, Dong Zhou, Ran Zmigrod, Markus Zopf.

Ethics Advisory Committee Reviewers:

Jade Abbott, Adewale Akinfaderin, Nora Al-Twairesh, Laura Alonso Alemany, David Alvarez-Melis, Maxime Amblard, Jean-Yves Antoine,

Timothy Baldwin, Kathy Baxter, Steven Bedrick, Luciana Benotti, Steven Bird, Claudia Borg, Jamie Brandon,

Kai-Wei Chang, Luis Chiruzzo, Marta R. Costa-jussà,

Guy Emerson,

Albert Gatt, Vasundhara Gautam, Dimitra Gkatzia, Sharon Goldwater, Alvin Grissom II,

Jack Hessel,

Shafiq Joty,

Anne Lauscher, Haley Lepp,

Nitin Madnani, Emiel van Miltenburg,

Aurélie Névéol, Nguyen Thi Minh Huyen,

José Ochoa-Luna,

Viviana Patti, Ted Pedersen,

Gabriela Ramírez-de-la-Rosa, Marta Recasens,

Tatjana Scheffler, Kathleen Siminyu,

Samson Tan, Rachael Tatman, Esaú Villatoro Tello.

Aline Villavicencio,

Kellie Webster, Richard Wicentowski,

Jingbo Xia.

Keynote Talk: Advancing Technological Equity in Speech and Language Processing

Helen Meng
The Chinese University of Hong Kong (CUHK)

Abstract: Accelerating advances in AI and deep neural networks have powered the proliferation of speech and language technologies in applications such as virtual assistants, smart speakers, reading machines, etc. The technologies have performed impressively well, achieving human parity in speech recognition accuracies and speech synthesis naturalness. As these technologies continue to permeate our daily lives, they need to support diverse users and usage contexts with inputs that deviate from the mainstream. Examples include non-native speakers, code-switching, speech carrying myriad emotions and styles, and speakers with impairments and disorders. Under such contexts, existing technologies often suffer performance degradations and fail to fulfill the needs of the users. The crux of the problem lies in data scarcity and data sparsity, which are exacerbated by high data variability.

This talk presents an overview of some of the approaches we have used to address the challenges of data shortage, positioned at various stages along the processing pipeline. They include: data augmentation based on speech signal perturbations, use of pre-trained representations, learning speech representation disentanglement, knowledge distillation architectures, meta-learned model re-initialization, as well as adversarially trained models. The effectiveness of these approaches are demonstrated through a variety of applications, including accented speech recognition, dysarthric speech recognition, code-switched speech synthesis, disordered speech reconstruction, one-shot voice conversion and exemplar-based emotive speech synthesis. These efforts strive to develop speech and language technologies that can gracefully adapt and accommodate a diversity of user needs and usage contexts, in order to achieve technological equity in our society.

Bio: Helen Meng is Patrick Huen Wing Ming Professor of Systems Engineering and Engineering Management at The Chinese University of Hong Kong (CUHK). Her research interests include speech and language technologies to support multilingual and multimodal human-computer interactions, eLearning and assistive technologies, as well as big data decision analytics using AI. She leads the interdisciplinary research team that received the first Theme-based Research Scheme Project in Artificial Intelligence in 2019 from the Hong Kong SAR Government's Research Grants Council. She is Chair of the Curriculum Development in the CUHK-JC AI4Future Project, which has developed the courseware for pre-tertiary AI education being taught in a growing number of participating secondary schools across Hong Kong.

Helen received all her degrees from MIT. She is the Founding Director of the CUHK Ministry of Education (MoE)-Microsoft Key Laboratory for Human-Centric Computing and Interface Technologies (since 2005), Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems (since 2006), and Stanley Ho Big Data Decision Analytics Research Center (since 2013). Previously, she has served as CUHK Faculty of Engineering's Associate Dean (Research), Chairman of the Department of Systems Engineering and Engineering Management, Editor-in-Chief of the IEEE Transactions on Audio, Speech and Language Processing, Member of the IEEE Signal Processing Society Board of Governors, ISCA Board Member and presently member of the IEEE SPS Awards Board and ISCA International Advisory Council. She was elected APSIPA's inaugural Distinguished Lecturer 2012-2013 and ISCA Distinguished Lecturer 2015-2016. Her awards include the Ministry of Education Higher Education Outstanding Scientific Research Output Award 2009, Microsoft Research Outstanding Collaborator Award 2016 (1 in 32 worldwide), IBM Faculty Award 2016, HKPWE Outstanding Women Professionals and Entrepreneurs Award 2017 (1 in 20 since 1999), Hong Kong ICT Silver Award 2018 in Smart Inclusion, 2019 IEEE SPS Leo L. Beranek Meritorious Service Award and various best paper

awards. Helen has served in a number of government appointments, which include memberships in the Steering Committee of Hong Kong's Electronic Health Record Sharing, Social Welfare Department's Joint Committee on Information Technology for the Social Welfare Sector and Advisory Committee on financing social welfare services. She is also a member of the AI4SDGs AI for Children Working Group. Helen is a Fellow of IEEE, ISCA, HKIE and HKCS.

Keynote Talk: Learning and Processing Language from Wearables: Opportunities and Challenges

Alejandrina Cristia
Laboratoire de Sciences Cognitives et de Psycholinguistique,
Département d'études cognitives, ENS, EHESS, CNRS, PSL University

Abstract: Recent years have seen tremendous improvement in the ease with which we can collect naturalistic language samples via devices worn over long periods of time. These allow unprecedented access to ego-centered experiences in language perceived and produced, including by young children. For example, in a newly-formed consortium, we pulled together over 40k hours of audio, collected from 1,001 children growing up in industrialized or hunter-horticulturalist populations, located in one of 12 countries. Such data are interesting for many purposes, including as 1. fodder for unsupervised language learning models aimed at mimicking what the child does; 2. indices of early language development that can be used to assess the impact of behavioral and pharmacological interventions; and 3. samples of the natural use of language(s) in low-resource and multilingual settings. The technology allowing to carve out interesting information from these large datasets, however, is lagging behind – but this may not be such a bad thing after all, since the ethical, technical, and legal handling of such data also need some work to increase the chances that the net impact of research based on this technique is positive. In this talk, I draw from cutting-edge research building on long-form recordings from wearables and a framework for doing the most good we can (effective altruism) to highlight surprising findings in early language acquisition, and delineate key priorities for future work.

Bio: Alejandrina Cristia is a senior researcher at the Centre National de la Recherche Scientifique (CNRS), leader of the Language Acquisition Across Cultures team, and director of the Laboratoire de Sciences Cognitives et Psycholinguistique (LSCP) cohosted by the Ecole Normale Supérieure, EHESS, and PSL. In 2021, she is an invited researcher in the Foundations of Learning Program of the Abdul Latif Jameel Poverty Action Lab (J-PAL), and a guest researcher at the Max Planck Institute for Evolutionary Anthropology. Her long-term aim is to answer the following questions: What are the linguistic representations that infants and adults have? Why and how are they formed? How may learnability biases shape the world's languages? To answer these questions, she combines multiple methodologies including spoken corpora analyses, behavioral studies, neuroimaging (NIRS), and computational modeling. This interdisciplinary approach has resulted in over 100 publications in pscyhology, linguistics, and development journals as well as IEEE and similar conferences. With an interest in cumulative, collaborative, and transparent science, she contributed to the creation of the first meta-meta-analysis platform (metalab.stanford.edu) and several international networks, including saliently the LangVIEW consortium that is leading /L+/, the First truly global summer/winter school on language acquisition.[1] She received the 2017 John S. McDonnell Scholar Award in Understanding Human Cognition, the 2020 Médaille de Bronze CNRS Section Linguistique, and an ERC Consolidator Award (2021-2026) for the ExELang[2] project.

[1] https://www.dpss.unipd.it/summer-school-2021/home
[2] exelang.fr

Keynote Talk: Reliable Characterizations of NLP Systems as a Social Responsibility

Christopher Potts
Stanford University

Abstract: This is an incredible moment for NLP. We all routinely work with models whose capabilities would have seemed like science fiction just two decades ago, powerful organizations eagerly await our latest results, and NLP technologies are playing an increasingly large role in shaping our society. As a result, all of us in the NLP community are likely to participate in research that will contribute (to varying degrees and perhaps only indirectly) to technologies that will impact many people's lives, with both positive and negative consequences – for example, technologies that broaden accessibility, enhance creative self-expression, heighten surveillance, and create propaganda. What can we do to fulfill the social responsibility that this brings? As a (very) partial answer to this question, I will review a number of important recent developments, spanning many research groups, concerning dataset creation, model introspection, and system assessment. Taken together, these ideas can help us more reliably characterize how NLP systems will behave, and more reliably communicate this information to a wider range of potential users. In this way, they can help us meet our obligations to the people whose lives are impacted by the results of our research.

Bio: Christopher Potts is Professor and Chair of Linguistics and Professor (by courtesy) of Computer Science at Stanford, and a faculty member in the Stanford NLP Group and the Stanford AI Lab. His group uses computational methods to explore how emotion is expressed in language and how linguistic production and interpretation are influenced by the context of utterance. This research combines methods from linguistics, cognitive psychology, and computer science, in the service of both scientific discovery and technology development. He was previously Chief Scientist at Roam Analytics, a start-up focused on applying NLP in healthcare and the life sciences (now Parexel AI Labs). He is a long-time Action Editor at TACL, a frequent Area Chair at ACL conferences, and currently an Ethics Committee co-chair for EMNLP 2021.

Table of Contents

Conference Program

Monday, August 2, 2021 (all times UTC+0)

08:15–08:35 ***Opening Session***

08:40–09:00 ***Presidential Address***

09:00–10:00 ***Keynote 1. Helen Meng: Advancing Technological Equity in Speech and Language Processing***

Session 1A: Computational Social Science and Cultural Analytics 1

10:00–10:10 *Investigating label suggestions for opinion mining in German Covid-19 social media*
Tilman Beck, Ji-Ung Lee, Christina Viehmann, Marcus Maurer, Oliver Quiring and Iryna Gurevych

10:10–10:20 *How Did This Get Funded?! Automatically Identifying Quirky Scientific Achievements*
Chen Shani, Nadav Borenstein and Dafna Shahaf

10:20–10:30 *Engage the Public: Poll Question Generation for Social Media Posts*
Zexin Lu, Keyang Ding, Yuji Zhang, Jing Li, Baolin Peng and Lemao Liu

10:30–10:40 *HateCheck: Functional Tests for Hate Speech Detection Models*
Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts and Janet Pierrehumbert

10:40–10:50 *Unified Dual-view Cognitive Model for Interpretable Claim Verification*
Lianwei Wu, Yuan Rao, Yuqian Lan, Ling Sun and Zhaoyin Qi

10:50–10:57 *Catchphrase: Automatic Detection of Cultural References*
Nir Sweed and Dafna Shahaf

Session 1B: Language Generation 1

10:00–10:10 *DeepRapper: Neural Rap Generation with Rhyme and Rhythm Modeling*
Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang and Tie-Yan Liu

10:10–10:20 *PENS: A Dataset and Generic Framework for Personalized News Headline Generation*
Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He and Xing Xie

10:20–10:30 *Enhancing Content Preservation in Text Style Transfer Using Reverse Attention and Conditional Layer Normalization*
Dongkyu Lee, Zhiliang Tian, Lanqing Xue and Nevin L. Zhang

10:30–10:40 *Mention Flags (MF): Constraining Transformer-based Text Generators*
Yufei Wang, Ian Wood, Stephen Wan, Mark Dras and Mark Johnson

10:40–10:50 *Generalising Multilingual Concept-to-Text NLG with Language Agnostic Delexicalisation*
Giulio Zhou and Gerasimos Lampouras

10:50–10:57 *On Training Instance Selection for Few-Shot Neural Text Generation*
Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh and Vera Demberg

Session 1C: Dialog and Interactive Systems 1

10:00–10:10 *Conversations Are Not Flat: Modeling the Dynamic Information Flow across Dialogue Utterances*
Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng and Jie Zhou

10:10–10:20 *Dual Slot Selector via Local Reliability Verification for Dialogue State Tracking*
Jinyu Guo, Kai Shuang, Jijie Li and Zihan Wang

10:20–10:30 *Transferable Dialogue Systems and User Simulators*
Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig and Bill Byrne

10:30–10:40 *BoB: BERT Over BERT for Training Persona-based Dialogue Models from Limited Personalized Data*
Haoyu Song, Yan Wang, Kaiyan Zhang, Wei-Nan Zhang and Ting Liu

10:40–10:50 *GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling*
Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che and Ting Liu

10:50–10:57 *Coreference Resolution without Span Representations*
Yuval Kirstain, Ori Ram and Omer Levy

Session 1D: Information Extraction 1

10:00–10:10 *Accelerating BERT Inference for Sequence Labeling via Early-Exit*
Xiaonan Li, Yunfan Shao, Tianxiang Sun, Hang Yan, Xipeng Qiu and Xuanjing Huang

10:10–10:20 *Modularized Interaction Network for Named Entity Recognition*
Fei Li, Zheng Wang, Siu Cheung Hui, Lejian Liao, Dandan Song, Jing Xu, Guoxiu He and meihuizi jia

10:20–10:30 *Capturing Event Argument Interaction via A Bi-Directional Entity-Level Recurrent Decoder*
Xi Xiangyu, Wei Ye, Shikun Zhang, Quanxiu Wang, Huixing Jiang and Wei Wu

10:30–10:40 *UniRE: A Unified Label Space for Entity Relation Extraction*
Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li and Junchi Yan

10:40–10:50 *Refining Sample Embeddings with Relation Prototypes to Enhance Continual Relation Extraction*
Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi and Yanghua Xiao

10:50–10:57 *Enhancing Entity Boundary Detection for Better Chinese Named Entity Recognition*
Chun Chen and Fang Kong

Monday, August 2, 2021 (all times UTC+0) (continued)

Session 1E: Machine Translation and Multilinguality 1

10:00–10:10 *Contrastive Learning for Many-to-many Multilingual Neural Machine Translation*
Xiao Pan, Mingxuan Wang, Liwei Wu and Lei Li

10:10–10:20 *Understanding the Properties of Minimum Bayes Risk Decoding in Neural Machine Translation*
Mathias Müller and Rico Sennrich

10:20–10:30 *Multi-Head Highly Parallelized LSTM Decoder for Neural Machine Translation*
Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong and Meng Zhang

10:30–10:40 *A Bidirectional Transformer Based Alignment Model for Unsupervised Word Alignment*
Jingyi Zhang and Josef van Genabith

10:40–10:50 *Learning Language Specific Sub-network for Multilingual Machine Translation*
Zehui Lin, Liwei Wu, Mingxuan Wang and Lei Li

10:50–10:57 *Difficulty-Aware Machine Translation Evaluation*
Runzhe Zhan, Xuebo Liu, Derek F. Wong and Lidia S. Chao

Session 2A: Sentiment Analysis, Stylistic Analysis, and Argument Mining 1

11:00–11:10 *Exploring the Efficacy of Automatically Generated Counterfactuals for Sentiment Analysis*
Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth and Ruihai Dong

11:10–11:20 *Bridge-Based Active Domain Adaptation for Aspect Term Extraction*
Zhuang Chen and Tieyun Qian

11:20–11:30 *Multimodal Sentiment Detection Based on Multi-channel Graph Neural Networks*
Xiaocui Yang, Shi Feng, Yifei Zhang and Daling Wang

11:30–11:40 *Aspect-Category-Opinion-Sentiment Quadruple Extraction with Implicit Aspects and Opinions*
Hongjie Cai, Rui Xia and Jianfei Yu

11:40–11:47 *Uncertainty and Surprisal Jointly Deliver the Punchline: Exploiting Incongruity-Based Features for Humor Recognition*
Yubo Xie, Junze Li and Pearl Pu

11:47–11:54 *Counterfactuals to Control Latent Disentangled Text Representations for Style Transfer*
Sharmila Reddy Nangi, Niyati Chhaya, Sopan Khosla, Nikhil Kaushik and Harshit Nyati

Session 2B: Summarization 1

11:00–11:10 *PASS: Perturb-and-Select Summarizer for Product Reviews*
Nadav Oved and Ran Levy

11:10–11:20 *Deep Differential Amplifier for Extractive Summarization*
Ruipeng Jia, Yanan Cao, Fang Fang, Yuchen Zhou, Zheng Fang, Yanbing Liu and Shi Wang

11:20–11:30 *Multi-TimeLine Summarization (MTLS): Improving Timeline Summarization by Generating Multiple Summaries*
Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama and Masatoshi Yoshikawa

11:30–11:40 *Self-Supervised Multimodal Opinion Summarization*
Jinbae Im, Moonki Kim, Hoyeop Lee, Hyunsouk Cho and Sehee Chung

11:40–11:50 *A Training-free and Reference-free Summarization Evaluation Metric via Centrality-weighted Relevance and Self-referenced Redundancy*
Wang Chen, Piji Li and Irwin King

11:50–12:00 *DESCGEN: A Distantly Supervised Datasetfor Generating Entity Descriptions*
Weijia Shi, Mandar Joshi and Luke Zettlemoyer

Session 2C: Interpretability and Analysis of Models for NLP 1

11:00–11:10 *Introducing Orthogonal Constraint in Structural Probes*
Tomasz Limisiewicz and David Mareček

11:10–11:20 *Hidden Killer: Invisible Textual Backdoor Attacks with Syntactic Trigger*
Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang and Maosong Sun

11:20–11:30 *Examining the Inductive Bias of Neural Language Models with Artificial Languages*
Jennifer C. White and Ryan Cotterell

11:30–11:40 *Explaining Contextualization in Language Models using Visual Analytics*
Rita Sevastjanova, Aikaterini-Lida Kalouli, Christin Beck, Hanna Schäfer and Mennatallah El-Assady

11:40–11:50 *Improving the Faithfulness of Attention-based Explanations with Task-specific Information for Text Classification*
George Chrysostomou and Nikolaos Aletras

11:50–11:57 *Attention Flows are Shapley Value Explanations*
Kawin Ethayarajh and Dan Jurafsky

Session 2D: Language Grounding to Vision, Robotics and Beyond 1

11:00–11:10 *Generating Landmark Navigation Instructions from Maps as a Graph-to-Text Problem*
Raphael Schumann and Stefan Riezler

11:10–11:20 *E2E-VLP: End-to-End Vision-Language Pre-training Enhanced by Visual Learning*
Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao and Fei Huang

11:20–11:30 *Learning Relation Alignment for Calibrated Cross-modal Retrieval*
Shuhuai Ren, Junyang Lin, Guangxiang Zhao, Rui Men, An Yang, Jingren Zhou, Xu Sun and Hongxia Yang

11:30–11:40 *KM-BART: Knowledge Enhanced Multimodal BART for Visual Commonsense Generation*
Yiran Xing, Zai Shi, Zhao Meng, Gerhard Lakemeyer, Yunpu Ma and Roger Wattenhofer

11:40–11:47 *Video Paragraph Captioning as a Text Summarization Task*
Hui Liu and Xiaojun Wan

11:47–11:54 *Are VQA Systems RAD? Measuring Robustness to Augmented Data with Focused Interventions*
Daniel Rosenberg, Itai Gat, Amir Feder and Roi Reichart

Session 2E: Machine Learning for NLP 1

11:00–11:10 *Cascaded Head-colliding Attention*
Lin Zheng, Zhiyong Wu and Lingpeng Kong

11:10–11:20 *Structural Knowledge Distillation: Tractably Distilling Information for Structured Predictor*
Xinyu Wang, Yong Jiang, Zhaohui Yan, Zixia Jia, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang and Kewei Tu

11:20–11:30 *Parameter-efficient Multi-task Fine-tuning for Transformers via Shared Hypernetworks*
Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani and James Henderson

11:30–11:40 *COSY: COunterfactual SYntax for Cross-Lingual Understanding*
SICHENG YU, Hao Zhang, Yulei Niu, Qianru Sun and Jing Jiang

11:40–11:50 *OoMMix: Out-of-manifold Regularization in Contextual Embedding Space for Text Classification*
Seonghyeon Lee, Dongha Lee and Hwanjo Yu

11:50–11:57 *How Helpful is Inverse Reinforcement Learning for Table-to-Text Generation?*
Sayan Ghosh, Zheng Qi, Snigdha Chaturvedi and Shashank Srivastava

Session 3A: Computational Social Science and Cultural Analytics 2

14:00–14:10 *Understanding and Countering Stereotypes: A Computational Approach to the Stereotype Content Model*
Kathleen C. Fraser, Isar Nejadgholi and Svetlana Kiritchenko

14:10–14:20 *Structurizing Misinformation Stories via Rationalizing Fact-Checks*
Shan Jiang and Christo Wilson

14:20–14:30 *Modeling Language Usage and Listener Engagement in Podcasts*
Sravana Reddy, Mariya Lazarova, Yongze Yu and Rosie Jones

14:30–14:40 *Breaking Down the Invisible Wall of Informal Fallacies in Online Discussions*
Saumya Sahai, Oana Balalau and Roxana Horincar

14:40–14:50 *SocAoG: Incremental Graph Parsing for Social Relation Inference in Dialogues*
Liang Qiu, Yuan Liang, Yizhou Zhao, Pan Lu, Baolin Peng, Zhou Yu, Ying Nian Wu and Song-Chun Zhu

14:50–14:57 *Automatic Fake News Detection: Are Models Learning to Reason?*
Casper Hansen, Christian Hansen and Lucas Chaves Lima

Session 3B: Dialog and Interactive Systems 2

14:00–14:10 *TicketTalk: Toward human-level performance with end-to-end, transaction-based dialog systems*
Bill Byrne, Karthik Krishnamoorthi, Saravanan Ganesh and Mihir Kale

14:10–14:20 *Improving Dialog Systems for Negotiation with Personality Modeling*
Runzhe Yang, Jingxiao Chen and Karthik Narasimhan

14:20–14:30 *Learning from Perturbations: Diverse and Informative Dialogue Generation with Inverse Adversarial Training*
Wangchunshu Zhou, Qifei LI and Chenle Li

14:30–14:40 *Increasing Faithfulness in Knowledge-Grounded Dialogue with Controllable Features*
Hannah Rashkin, David Reitter, Gaurav Singh Tomar and Dipanjan Das

14:40–14:47 *Saying No is An Art: Contextualized Fallback Responses for Unanswerable Dialogue Queries*
Ashish Shrivastava, Kaustubh Dhole, Abhinav Bhatt and Sharvani Raghunath

14:47–14:54 *N-Best ASR Transformer: Enhancing SLU Performance using Multiple ASR Hypotheses*
Karthik Ganesan, Pakhi Bamdev, Jaivarsan B, Amresh Venugopal and Abhinav Tushar

Session 3C: Information Extraction 2

14:00–14:10 *CitationIE: Leveraging the Citation Graph for Scientific Information Extraction*
Vijay Viswanathan, Graham Neubig and Pengfei Liu

14:10–14:20 *From Discourse to Narrative: Knowledge Projection for Event Relation Extraction*
Jialong Tang, Hongyu Lin, Meng Liao, Yaojie Lu, Xianpei Han, Le Sun, Weijian Xie and Jin Xu

14:20–14:30 *AdvPicker: Effectively Leveraging Unlabeled Data via Adversarial Discriminator for Cross-Lingual NER*
Weile Chen, Huiqiang Jiang, Qianhui Wu, Börje Karlsson and Yi Guan

14:30–14:40 *Compare to The Knowledge: Graph Neural Fake News Detection with External Knowledge*
Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan and Ming Zhou

14:40–14:50 *Discontinuous Named Entity Recognition as Maximal Clique Discovery*
Yucheng Wang, Bowen Yu, Hongsong Zhu, Tingwen Liu, Nan Yu and Limin Sun

14:50–15:00 *LNN-EL: A Neuro-Symbolic Approach to Short-text Entity Linking*
Hang Jiang, Sairam Gurajada, Qiuhao Lu, Sumit Neelam, Lucian Popa, Prithviraj Sen, Yunyao Li and Alexander Gray

Session 3D: Machine Translation and Multilinguality 2

14:00–14:10 *Do Context-Aware Translation Models Pay the Right Attention?*
Kayo Yin, Patrick Fernandes, Danish Pruthi, Aditi Chaudhary, André F. T. Martins and Graham Neubig

14:10–14:20 *Adapting High-resource NMT Models to Translate Low-resource Related Languages without Parallel Data*
Wei-Jen Ko, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Naman Goyal, Francisco Guzmán, Pascale Fung, Philipp Koehn and Mona Diab

14:20–14:30 *Bilingual Lexicon Induction via Unsupervised Bitext Construction and Word Alignment*
Haoyue Shi, Luke Zettlemoyer and Sida I. Wang

14:30–14:40 *Multilingual Speech Translation from Efficient Finetuning of Pretrained Models*
Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau and Michael Auli

14:40–14:47 *Gender bias amplification during Speed-Quality optimization in Neural Machine Translation*
Adithya Renduchintala, Denise Diaz, Kenneth Heafield, Xian Li and Mona Diab

14:47–14:54 *Machine Translation into Low-resource Language Varieties*
Sachin Kumar, Antonios Anastasopoulos, Shuly Wintner and Yulia Tsvetkov

Session 3E: Interpretability and Analysis of Models for NLP 2

14:00–14:10 *Learning Faithful Representations of Causal Graphs*
Ananth Balashankar and Lakshminarayanan Subramanian

14:10–14:20 *What Context Features Can Transformer Language Models Use?*
Joe O'Connor and Jacob Andreas

14:20–14:30 *Integrated Directional Gradients: Feature Interaction Attribution for Neural NLP Models*
Sandipan Sikdar, Parantapa Bhattacharya and Kieran Heese

14:30–14:37 *Is Sparse Attention more Interpretable?*
Clara Meister, Stefan Lazov, Isabelle Augenstein and Ryan Cotterell

Monday, August 2, 2021 (all times UTC+0) (continued)

14:37–14:44 *The Case for Translation-Invariant Self-Attention in Transformer-Based Language Models*
Ulme Wennberg and Gustav Eje Henter

14:44–14:51 *Relative Importance in Sentence Processing*
Nora Hollenstein and Lisa Beinborn

Poster 1A: Semantics: Sentence-level Semantics, Textual Inference and Other areas

15:00–17:00 *DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations*
John Giorgi, Osvald Nitski, Bo Wang and Gary Bader

15:00–17:00 *Doing Good or Doing Right? Exploring the Weakness of Commonsense Causal Reasoning Models*
Mingyue Han and Yinglin Wang

15:00–17:00 *XLPT-AMR: Cross-Lingual Pre-Training via Multi-Task Learning for Zero-Shot AMR Parsing and Text Generation*
Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang and Guodong Zhou

15:00–17:00 *Span-based Semantic Parsing for Compositional Generalization*
Jonathan Herzig and Jonathan Berant

15:00–17:00 *AND does not mean OR: Using Formal Languages to Study Language Models' Representations*
Aaron Traylor, Roman Feiman and Ellie Pavlick

15:00–17:00 *Enforcing Consistency in Weakly Supervised Semantic Parsing*
Nitish Gupta, Sameer Singh and Matt Gardner

15:00–17:00 *Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both?*
Peter Shaw, Ming-Wei Chang, Panupong Pasupat and Kristina Toutanova

Poster 1B: Linguistic Theories, Cognitive Modeling and Psycholinguistics

15:00–17:00 *A Targeted Assessment of Incremental Processing in Neural Language Models and Humans*
Ethan Wilcox, Pranali Vani and Roger Levy

Poster 1C: Semantics: Lexical Semantics

15:00–17:00 *The Possible, the Plausible, and the Desirable: Event-Based Modality Detection for Language Processing*
Valentina Pyatkin, Shoval Sadde, Aynat Rubinstein, Paul Portner and Reut Tsarfaty

Poster 1D: Phonology, Morphology and Word Segmentation

15:00–17:00 *To POS Tag or Not to POS Tag: The Impact of POS Tags on Morphological Learning in Low-Resource Settings*
Sarah Moeller, Ling Liu and Mans Hulden

Poster 1E: Speech and Multimodality

15:00–17:00 *Prosodic segmentation for parsing spoken dialogue*
Elizabeth Nielsen, Mark Steedman and Sharon Goldwater

15:00–17:00 *VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation*
Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino and Emmanuel Dupoux

15:00–17:00 *An Improved Model for Voicing Silent Speech*
David Gaddy and Dan Klein

Poster 1F: Ethics in NLP

15:00–17:00 *What's in the Box? An Analysis of Undesirable Content in the Common Crawl Corpus*
Alexandra Luccioni and Joseph Viviano

15:00–17:00 *Stereotyping Norwegian Salmon: An Inventory of Pitfalls in Fairness Benchmark Datasets*
Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim and Hanna Wallach

Poster 1G: Information Retrieval and Text Mining

15:00–17:00 *Robust Knowledge Graph Completion with Stacked Convolutions and a Student Re-Ranking Network*
Justin Lovelace, Denis Newman-Griffis, Shikhar Vashishth, Jill Fain Lehman and Carolyn Rosé

15:00–17:00 *A DQN-based Approach to Finding Precise Evidences for Fact Verification*
Hai Wan, Haicheng Chen, Jianfeng Du, Weilin Luo and Rongzhen Ye

Poster 1H: Machine Learning for NLP

15:00–17:00 *The Art of Abstention: Selective Prediction and Error Regularization for Natural Language Processing*
Ji Xin, Raphael Tang, Yaoliang Yu and Jimmy Lin

15:00–17:00 *Unsupervised Out-of-Domain Detection via Pre-trained Transformers*
Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng and Caiming Xiong

15:00–17:00 *Continual Quality Estimation with Online Bayesian Meta-Learning*
Abiola Obamuyide, Marina Fomicheva and Lucia Specia

15:00–17:00 *MATE-KD: Masked Adversarial TExt, a Companion to Knowledge Distillation*
Ahmad Rashid, Vasileios Lioutas and Mehdi Rezagholizadeh

15:00–17:00 *Selecting Informative Contexts Improves Language Model Fine-tuning*
Richard Antonello, Nicole Beckage, Javier Turek and Alexander Huth

15:00–17:00 *Explainable Prediction of Text Complexity: The Missing Preliminaries for Text Simplification*
Cristina Garbacea, Mengtian Guo, Samuel Carton and Qiaozhu Mei

15:00–17:00 *Multi-Task Retrieval for Knowledge-Intensive Tasks*
Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oguz, Veselin Stoyanov and Gargi Ghosh

Poster 1I: Interpretability and Analysis of Models for NLP

15:00–17:00 *When Do You Need Billions of Words of Pretraining Data?*
Yian Zhang, Alex Warstadt, Xiaocheng Li and Samuel R. Bowman

15:00–17:00 *Analyzing the Source and Target Contributions to Predictions in Neural Machine Translation*
Elena Voita, Rico Sennrich and Ivan Titov

15:00–17:00 *Comparing Test Sets with Item Response Theory*
Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho and Samuel R. Bowman

15:00–17:00 *Uncovering Constraint-Based Behavior in Neural Models via Targeted Fine-Tuning*
Forrest Davis and Marten van Schijndel

15:00–17:00 *More Identifiable yet Equally Performant Transformers for Text Classification*
Rishabh Bhardwaj, Navonil Majumder, Soujanya Poria and Eduard Hovy

Poster 1J: Dialog and Interactive Systems

15:00–17:00 *AugNLG: Few-shot Natural Language Generation using Self-trained Data Augmentation*
Xinnuo Xu, Guoyin Wang, Young-Bum Kim and Sungjin Lee

15:00–17:00 *A Span-based Dynamic Local Attention Model for Sequential Sentence Classification*
Xichen Shang, Qianli Ma, Zhenxi Lin, Jiangyue Yan and Zipeng Chen

Poster 1K: Resources and Evaluation

15:00–17:00 *How effective is BERT without word ordering? Implications for language understanding and data privacy*
Jack Hessel and Alexandra Schofield

15:00–17:00 *Can vectors read minds better than experts? Comparing data augmentation strategies for the automated scoring of children's mindreading ability*
Venelin Kovatchev, Phillip Smith, Mark Lee and Rory Devine

15:00–17:00 *A Dataset and Baselines for Multilingual Reply Suggestion*
Mozhi Zhang, Wei Wang, Budhaditya Deb, Guoqing Zheng, Milad Shokouhi and Ahmed Hassan Awadallah

15:00–17:00 *WikiSum: Coherent Summarization Dataset for Efficient Human-Evaluation*
Nachshon Cohen, Oren Kalinsky, Yftah Ziser and Alessandro Moschitti

15:00–17:00 *What Ingredients Make for an Effective Crowdsourcing Protocol for Difficult NLU Data Collection Tasks?*
Nikita Nangia, Saku Sugawara, Harsh Trivedi, Alex Warstadt, Clara Vania and Samuel R. Bowman

15:00–17:00 *UMIC: An Unreferenced Metric for Image Captioning via Contrastive Learning*
Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Trung Bui and Kyomin Jung

15:00–17:00 *Neural OCR Post-Hoc Correction of Historical Corpora*
Lijun Lyu, Maria Koutraki, Martin Krikl and Besnik Fetahu

Poster 1L: Computational Social Science and Cultural Analytics

15:00–17:00 *Align Voting Behavior with Public Statements for Legislator Representation Learning*
Xinyi Mou, Zhongyu Wei, Lei Chen, Shangyi Ning, Yancheng He, Changjian Jiang and Xuanjing Huang

15:00–17:00 *Measure and Evaluation of Semantic Divergence across Two Languages*
Syrielle Montariol and Alexandre Allauzen

Poster 1M: Machine Translation and Multilinguality

15:00–17:00 *Improving Zero-Shot Translation by Disentangling Positional Information*
Danni Liu, Jan Niehues, James Cross, Francisco Guzmán and Xian Li

15:00–17:00 *Common Sense Beyond English: Evaluating and Improving Multilingual Language Models for Commonsense Reasoning*
Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao and Xiang Ren

15:00–17:00 *Attention Calibration for Transformer in Neural Machine Translation*
Yu Lu, Jiali Zeng, Jiajun Zhang, Shuangzhi Wu and Mu Li

15:00–17:00 *Anchor-based Bilingual Word Embeddings for Low-Resource Languages*
Tobias Eder, Viktor Hangya and Alexander Fraser

15:00–17:00 *Diverse Pretrained Context Encodings Improve Document Translation*
Domenic Donato, Lei Yu and Chris Dyer

15:00–17:00 *Multilingual Agreement for Multilingual Neural Machine Translation*
Jian Yang, Yuwei Yin, Shuming Ma, Haoyang Huang, Dongdong Zhang, Zhoujun Li and Furu Wei

15:00–17:00 *Exploiting Language Relatedness for Low Web-Resource Language Model Adaptation: An Indic Languages Study*
Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar and Sunita Sarawagi

Poster 1N: Syntax: Tagging, Chunking, and Parsing

15:00–17:00 *On Finding the K-best Non-projective Dependency Trees*
Ran Zmigrod, Tim Vieira and Ryan Cotterell

15:00–17:00 *Higher-order Derivatives of Weighted Finite-state Machines*
Ran Zmigrod, Tim Vieira and Ryan Cotterell

Poster 1O: Theme

15:00–17:00 *Towards Argument Mining for Social Good: A Survey*
Eva Maria Vecchi, Neele Falk, Iman Jundi and Gabriella Lapesa

15:00–17:00 *Automated Generation of Storytelling Vocabulary from Photographs for use in AAC*
Mauricio Fontana de Vargas and Karyn Moffatt

Poster 1P: NLP Applications

15:00–17:00 *CLIP: A Dataset for Extracting Action Items for Physicians from Hospital Discharge Notes*
James Mullenbach, Yada Pruksachatkun, Sean Adler, Jennifer Seale, Jordan Swartz, Greg McKelvey, Hui Dai, Yi Yang and David Sontag

15:00–17:00 *Assessing Emoji Use in Modern Text Processing Tools*
Abu Awal Md Shoeb and Gerard de Melo

15:00–17:00 *Select, Extract and Generate: Neural Keyphrase Generation with Layer-wise Coverage Attention*
Wasi Ahmad, Xiao Bai, Soomin Lee and Kai-Wei Chang

Poster 1Q: Language Generation

15:00–17:00 *Factorising Meaning and Form for Intent-Preserving Paraphrasing*
Tom Hosking and Mirella Lapata

15:00–17:00 *AggGen: Ordering and Aggregating while Generating*
Xinnuo Xu, Ondřej Dušek, Verena Rieser and Ioannis Konstas

15:00–17:00 *Reflective Decoding: Beyond Unidirectional Generation with Off-the-Shelf Language Models*
Peter West, Ximing Lu, Ari Holtzman, Chandra Bhagavatula, Jena D. Hwang and Yejin Choi

15:00–17:00 *Towards Table-to-Text Generation with Numerical Reasoning*
Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura and Hiroya Takamura

15:00–17:00 *Data-to-text Generation with Macro Planning*
Ratish Puduppully and Mirella Lapata

Poster 1R: Summarization

15:00–17:00 *BACO: A Background Knowledge- and Content-Based Framework for Citing Sentence Generation*
Yubin Ge, Ly Dinh, Xiaofeng Liu, Jinsong Su, Ziyao Lu, Ante Wang and Jana Diesner

15:00–17:00 *Language Model as an Annotator: Exploring DialoGPT for Dialogue Summarization*
Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin and Ting Liu

15:00–17:00 *Reinforcement Learning for Abstractive Question Summarization with Question-aware Semantic Rewards*
Shweta Yadav, Deepak Gupta, Asma Ben Abacha and Dina Demner-Fushman

Poster 1S: Question Answering

15:00–17:00 *Challenges in Information-Seeking QA: Unanswerable Questions and Paragraph Retrieval*
Akari Asai and Eunsol Choi

15:00–17:00 *A Semantics-aware Transformer Model of Relation Linking for Knowledge Base Question Answering*
Tahira Naseem, Srinivas Ravishankar, Nandana Mihindukulasooriya, Ibrahim Abdelaziz, Young-Suk Lee, Pavan Kapanipathi, Salim Roukos, Alfio Gliozzo and Alexander Gray

15:00–17:00 *A Gradually Soft Multi-Task and Data-Augmented Approach to Medical Question Understanding*
Khalil Mrini, Franck Dernoncourt, Seunghyun Yoon, Trung Bui, Walter Chang, Emilia Farcas and Ndapa Nakashole

15:00–17:00 *Neural Retrieval for Question Answering with Cross-Attention Supervised Data Augmentation*
Yinfei Yang, Ning Jin, Kuo Lin, Mandy Guo and Daniel Cer

Poster 1T: Language Grounding to Vision, Robotics and Beyond

15:00–17:00 *Enhancing Descriptive Image Captioning with Natural Language Inference*
Zhan Shi, Hui Liu and Xiaodan Zhu

Poster 1U: Information Extraction

15:00–17:00 *Leveraging Type Descriptions for Zero-shot Named Entity Recognition and Classification*
Rami Aly, Andreas Vlachos and Ryan McDonald

15:00–17:00 *MECT: Multi-Metadata Embedding based Cross-Transformer for Chinese Named Entity Recognition*
Shuang Wu, Xiaoning Song and Zhenhua Feng

15:00–17:00 *MOLEMAN: Mention-Only Linking of Entities with a Mention Annotation Network*
Nicholas FitzGerald, Dan Bikel, Jan Botha, Daniel Gillick, Tom Kwiatkowski and Andrew McCallum

15:00–17:00 *Factuality Assessment as Modal Dependency Parsing*
Jiarui Yao, Haoling Qiu, Jin Zhao, Bonan Min and Nianwen Xue

Poster 1V: Sentiment Analysis, Stylistic Analysis, and Argument Mining

15:00–17:00 *Directed Acyclic Graph Network for Conversational Emotion Recognition*
Weizhou Shen, Siyue Wu, Yunyi Yang and Xiaojun Quan

15:00–17:00 *Improving Formality Style Transfer with Context-Aware Rule Injection*
Zonghai Yao and hong yu

15:00–17:00 *Topic-Driven and Knowledge-Aware Transformer for Dialogue Emotion Detection*
Lixing Zhu, Gabriele Pergola, Lin Gui, Deyu Zhou and Yulan He

15:00–17:00 *Syntopical Graphs for Computational Argumentation Tasks*
Joe Barrow, Rajiv Jain, Nedim Lipka, Franck Dernoncourt, Vlad Morariu, Varun Manjunatha, Douglas Oard, Philip Resnik and Henning Wachsmuth

15:00–17:00 *Stance Detection in COVID-19 Tweets*
Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea and Cornelia Caragea

15:00–17:00 *eMLM: A New Pre-training Objective for Emotion Related Tasks*
Tiberiu Sosea and Cornelia Caragea

15:00–17:00 *Topic-Aware Evidence Reasoning and Stance-Aware Aggregation for Fact Verification*
Jiasheng Si, Deyu Zhou, Tongzhe Li, Xingyu Shi and Yulan He

17:00—18:00 ***Keynote 2. Alejandrina Cristia: Learning and Processing Language from Wearables: Opportunities and Challenges***

Session 4A: Computational Social Science and Cultural Analytics 3

23:00–23:10 *Changes in European Solidarity Before and During COVID-19: Evidence from a Large Crowd- and Expert-Annotated Twitter Dataset*
Alexandra Ils, Dan Liu, Daniela Grunow and Steffen Eger

23:10–23:20 *Measuring Conversational Uptake: A Case Study on Student-Teacher Interactions*
Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky and Tatsunori Hashimoto

23:20–23:30 *A Survey of Code-switching: Linguistic and Social Perspectives for Language Technologies*
A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock and Almedia Jacqueline Toribio

23:30–23:40 *Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection*
Bertie Vidgen, Tristan Thrush, Zeerak Waseem and Douwe Kiela

23:40–23:50 *InfoSurgeon: Cross-Media Fine-grained Information Consistency Checking for Fake News Detection*
Yi Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen McKeown, Mohit Bansal and Avi Sil

23:50–23:57 *On Positivity Bias in Negative Reviews*
Madhusudhan Aithal and Chenhao Tan

Session 4B: Dialog and Interactive Systems 3

23:00–23:10 *I like fish, especially dolphins: Addressing Contradictions in Dialogue Modeling*
Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela and Jason Weston

23:10–23:20 *A Sequence-to-Sequence Approach to Dialogue State Tracking*
Yue Feng, Yang Wang and Hang Li

23:20–23:30 *Discovering Dialog Structure Graph for Coherent Dialog Generation*
Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu and Wanxiang Che

23:30–23:40 *Dialogue Response Selection with Hierarchical Curriculum Learning*
Yixuan Su, Deng Cai, Qingyu Zhou, Zibo Lin, Simon Baker, Yunbo Cao, Shuming Shi, Nigel Collier and Yan Wang

Monday, August 2, 2021 (all times UTC+0) (continued)

23:40–23:50 *A Joint Model for Dropped Pronoun Recovery and Conversational Discourse Parsing in Chinese Conversational Speech*
Jingxuan Yang, Kerui Xu, Jun Xu, Si Li, Sheng Gao, Jun Guo, Nianwen Xue and Ji-Rong Wen

23:50–23:57 *PRAL: A Tailored Pre-Training Model for Task-Oriented Dialog Generation*
Jing Gu, Qingyang Wu, Chongruo Wu, Weiyan Shi and Zhou Yu

Session 4C: Information Extraction 3

23:00–23:10 *A Systematic Investigation of KB-Text Embedding Alignment at Scale*
Vardaan Pahuja, Yu Gu, Wenhu Chen, Mehdi Bahrami, Lei Liu, Wei-Peng Chen and Yu Su

23:10–23:20 *Named Entity Recognition with Small Strongly Labeled and Large Weakly Labeled Data*
Haoming Jiang, Danqing Zhang, Tianyu Cao, Bing Yin and Tuo Zhao

23:20–23:30 *Ultra-Fine Entity Typing with Weak Supervision from a Masked Language Model*
Hongliang Dai, Yangqiu Song and Haixun Wang

23:30–23:40 *Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning*
Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang and Kewei Tu

23:40–23:47 *ROPE: Reading Order Equivariant Positional Encoding for Graph-based Document Information Extraction*
Chen-Yu Lee, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat and Tomas Pfister

23:47–23:54 *Zero-shot Event Extraction via Transfer Learning: Challenges and Insights*
Qing Lyu, Hongming Zhang, Elior Sulem and Dan Roth

Session 4D: Interpretability and Analysis of Models for NLP 3

23:00–23:10 *Implicit Representations of Meaning in Neural Language Models*
Belinda Z. Li, Maxwell Nye and Jacob Andreas

23:10–23:20 *Causal Analysis of Syntactic Agreement Mechanisms in Neural Language Models*
Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen and Yonatan Belinkov

23:20–23:30 *Bird's Eye: Probing for Linguistic Graph Structures with a Simple Information-Theoretic Approach*
Yifan Hou and Mrinmaya Sachan

23:30–23:40 *Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases*
Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue and Jin Xu

23:40–23:50 *Poisoning Knowledge Graph Embeddings via Relation Inference Patterns*
Peru Bhardwaj, John Kelleher, Luca Costabello and Declan O'Sullivan

23:50–23:57 *Using Adversarial Attacks to Reveal the Statistical Bias in Machine Reading Comprehension Models*
Jieyu Lin, Jiajie Zou and Nai Ding

Session 4E: Ethics in NLP 1

23:00–23:10 *Bad Seeds: Evaluating Lexical Methods for Bias Measurement*
Maria Antoniak and David Mimno

23:10–23:20 *A Survey of Race, Racism, and Anti-Racism in NLP*
Anjalie Field, Su Lin Blodgett, Zeerak Waseem and Yulia Tsvetkov

23:20–23:30 *Intrinsic Bias Metrics Do Not Correlate with Application Bias*
Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya and Adam Lopez

23:30–23:40 *RedditBias: A Real-World Resource for Bias Evaluation and Debiasing of Conversational Language Models*
Soumya Barikeri, Anne Lauscher, Ivan Vulić and Goran Glavaš

Monday, August 2, 2021 (all times UTC+0) (continued)

23:40–23:47 *Quantifying and Avoiding Unfair Qualification Labour in Crowdsourcing*
Jonathan K. Kummerfeld

23:47–23:54 *Men Are Elected, Women Are Married: Events Gender Bias on Wikipedia*
Jiao Sun and Nanyun Peng

Tuesday, August 3, 2021 (all times UTC+0)

Session 5A: Machine Translation and Multilinguality 3

00:00–00:10 *Contributions of Transformer Attention Heads in Multi- and Cross-lingual Tasks*
Weicheng Ma, Kai Zhang, Renze Lou, Lili Wang and Soroush Vosoughi

00:10–00:20 *Crafting Adversarial Examples for Neural Machine Translation*
Xinze Zhang, Junzhe Zhang, Zhenhua Chen and Kun He

00:20–00:30 *UXLA: A Robust Unsupervised Data Augmentation Framework for Zero-Resource Cross-Lingual NLP*
M Saiful Bari, Tasnim Mohiuddin and Shafiq Joty

00:30–00:40 *Glancing Transformer for Non-Autoregressive Neural Machine Translation*
Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu and Lei Li

00:40–00:47 *Modeling Task-Aware MIMO Cardinality for Efficient Multilingual Neural Machine Translation*
Hongfei Xu, Qiuhui Liu, Josef van Genabith and Deyi Xiong

00:47–00:54 *Adaptive Nearest Neighbor Machine Translation*
Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo and Jiajun CHEN

Session 5B: Language Grounding to Vision, Robotics and Beyond 2

00:00–00:10 *Hierarchical Context-aware Network for Dense Video Event Captioning*
Lei Ji, Xianglin Guo, Haoyang Huang and Xilin Chen

00:10–00:20 *Control Image Captioning Spatially and Temporally*
Kun Yan, Lei Ji, Huaishao Luo, Ming Zhou, Nan Duan and Shuai Ma

00:20–00:30 *Edited Media Understanding Frames: Reasoning About the Intent and Implications of Visual Misinformation*
Jeff Da, Maxwell Forbes, Rowan Zellers, Anthony Zheng, Jena D. Hwang, Antoine Bosselut and Yejin Choi

00:30–00:40 *PIGLeT: Language Grounding Through Neuro-Symbolic Interaction in a 3D World*
Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi and Yejin Choi

00:40–00:50 *Neural Event Semantics for Grounded Language Understanding*
Shyamal Buch, Li Fei-Fei and Noah Goodman

Session 5C: Machine Learning for NLP 2

00:00–00:10 *Modeling Fine-Grained Entity Types with Box Embeddings*
Yasumasa Onoe, Michael Boratko, Andrew McCallum and Greg Durrett

00:10–00:20 *ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information*
zijun sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu and Jiwei Li

00:20–00:30 *Weight Distillation: Transferring the Knowledge in Neural Network Parameters*
Ye Lin, Yanyang Li, Ziyang Wang, Bei Li, Quan Du, Tong Xiao and Jingbo Zhu

00:30–00:40 *Optimizing Deeper Transformers on Small Datasets*
Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince and Yanshuai Cao

00:40–00:50 *BERTAC: Enhancing Transformer-based Language Models with Adversarially Pretrained Convolutional Neural Networks*
Jong-Hoon Oh, Ryu Iida, Julien Kloetzer and Kentaro Torisawa

00:50–00:57 *On Orthogonality Constraints for Transformers*
Aston Zhang, Alvin Chan, Yi Tay, Jie Fu, Shuohang Wang, Shuai Zhang, Huajie Shao, Shuochao Yao and Roy Ka-Wei Lee

Session 5D: NLP Applications 1 and Ethics

00:00–00:10 *COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic*
Arkadiy Saakyan, Tuhin Chakrabarty and Smaranda Muresan

00:10–00:20 *Explaining Relationships Between Scientific Documents*
Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola and Noah A. Smith

00:20–00:30 *IrEne: Interpretable Energy Prediction for Transformers*
Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian and Niranjan Balasubramanian

00:30–00:40 *Mitigating Bias in Session-based Cyberbullying Detection: A Non-Compromising Approach*
Lu Cheng, Ahmadreza Mosallanezhad, Yasin Silva, Deborah Hall and Huan Liu

00:40–00:50 *PlotCoder: Hierarchical Decoding for Synthesizing Visualization Code in Programmatic Context*
Xinyun Chen, Linyuan Gong, Alvin Cheung and Dawn Song

00:50–01:00 *Changing the World by Changing the Data*
Anna Rogers

Session 6A: Machine Learning for NLP 3

01:00–01:10 *EarlyBERT: Efficient BERT Training via Early-bird Lottery Tickets*
Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang and Jingjing Liu

01:10–01:20 *On the Effectiveness of Adapter-based Tuning for Pretrained Language Model Adaptation*
Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, BOSHENG DING, Liying Cheng, Jiawei Low, Lidong Bing and Luo Si

01:20–01:30 *Data Augmentation for Text Generation Without Any Augmented Data*
Wei Bi, Huayang Li and Jiacheng Huang

01:30–01:40 *KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation*
Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li and Jian Tang

01:40–01:50 *Integrating Semantics and Neighborhood Information with Graph-Driven Generative Models for Document Retrieval*
Zijing Ou, Qinliang Su, Jianxing Yu, Bang Liu, Jingwen Wang, Ruihui Zhao, Changyou Chen and Yefeng Zheng

01:50–01:57 *Measuring and Improving BERT's Mathematical Abilities by Predicting the Order of Reasoning.*
Piotr Piękos, Mateusz Malinowski and Henryk Michalewski

Session 6B: Resources and Evaluation 1

01:00–01:10 *SMURF: SeMantic and linguistic UndeRstanding Fusion for Caption Evaluation via Typicality Analysis*
Joshua Feinglass and Yezhou Yang

01:10–01:20 *KaggleDBQA: Realistic Evaluation of Text-to-SQL Parsers*
Chia-Hsuan Lee, Oleksandr Polozov and Matthew Richardson

01:20–01:30 *QASR: QCRI Aljazeera Speech Resource A Large Scale Annotated Arabic Speech Corpus*
Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury and Ahmed Ali

01:30–01:40 *An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models*
Xueqing Liu and Chi Wang

01:40–01:50 *Better than Average: Paired Evaluation of NLP systems*
Maxime Peyrard, Wei Zhao, Steffen Eger and Robert West

01:50–01:57 *Happy Dance, Slow Clap: Using Reaction GIFs to Predict Induced Affect on Twitter*
Boaz Shmueli, Soumya Ray and Lun-Wei Ku

Session 6C: Semantics: Sentence-level Semantics, Textual Inference and Other areas 1

01:00–01:10 *Chase: A Large-Scale and Pragmatic Chinese Dataset for Cross-Database Context-Dependent Text-to-SQL*
Jiaqi Guo, Ziliang Si, Yu Wang, Qian Liu, Ming Fan, Jian-Guang LOU, Zijiang Yang and Ting Liu

01:10–01:20 *CLINE: Contrastive Learning with Semantic Negative Examples for Natural Language Understanding*
Dong Wang, Ning Ding, Piji Li and Haitao Zheng

01:20–01:30 *Tree-Structured Topic Modeling with Nonparametric Neural Variational Inference*
Ziye Chen, Cheng Ding, Zusheng Zhang, Yanghui Rao and Haoran Xie

01:30–01:40 *ExCAR: Event Graph Knowledge Enhanced Explainable Causal Reasoning*
Li Du, Xiao Ding, Kai Xiong, Ting Liu and Bing Qin

01:40–01:50 *Infusing Finetuning with Semantic Dependencies*
Zhaofeng Wu, Hao Peng and Noah Smith

01:50–01:57 *Exploring Listwise Evidence Reasoning with T5 for Fact Verification*
Kelvin Jiang, Ronak Pradeep and Jimmy Lin

Session 6D: Sentiment Analysis, Stylistic Analysis, and Argument Mining 2

01:00–01:10 *Distributed Representations of Emotion Categories in Emotion Space*
Xiangyu Wang and Chengqing Zong

01:10–01:20 *Style is NOT a single variable: Case Studies for Cross-Stylistic Language Understanding*
Dongyeop Kang and Eduard Hovy

01:20–01:30 *DynaSent: A Dynamic Benchmark for Sentiment Analysis*
Christopher Potts, Zhengxuan Wu, Atticus Geiger and Douwe Kiela

01:30–01:40 *A Hierarchical VAE for Calibrating Attributes while Generating Text using Normalizing Flow*
Bidisha Samanta, Mohit Agrawal and NIloy Ganguly

01:40–01:50 *A Unified Generative Framework for Aspect-based Sentiment Analysis*
Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu and Zheng Zhang

01:50–02:00 *Classifying Argumentative Relations Using Logical Mechanisms and Argumentation Schemes*
Yohan Jo, Seojin Bang, Chris Reed and Eduard Hovy

Session 7A: Dialog and Interactive Systems 4

08:00–08:10 *Discovering Dialogue Slots with Weak Supervision*
Vojtěch Hudeček, Ondřej Dušek and Zhou Yu

08:10–08:20 *Enhancing the generalization for Intent Classification and Out-of-Domain Detection in SLU*
Yilin Shen, Yen-Chang Hsu, Avik Ray and Hongxia Jin

08:20–08:30 *ProtAugment: Intent Detection Meta-Learning through Unsupervised Diverse Paraphrasing*
Thomas Dopierre, Christophe Gravier and Wilfried Logerais

08:30–08:40 *Robustness Testing of Language Understanding in Task-Oriented Dialog*
Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, hongguang li, weiran nie, Cheng LI, Wei Peng and Minlie Huang

08:40–08:50 *Comprehensive Study: How the Context Information of Different Granularity Affects Dialogue State Tracking?*
Puhai Yang, Heyan Huang and Xian-Ling Mao

08:50–09:00 *OTTers: One-turn Topic Transitions for Open-Domain Dialogue*
Karin Sevegnani, David M. Howcroft, Ioannis Konstas and Verena Rieser

Session 7B: Semantics: Sentence-level Semantics, Textual Inference and Other areas 2

08:00–08:10 *Towards Robustness of Text-to-SQL Models against Synonym Substitution*
Yujian Gan, Xinyun Chen, Qiuping Huang, Matthew Purver, John R. Woodward, Jinxia Xie and Pengsheng Huang

08:10–08:20 *KACE: Generating Knowledge Aware Contrastive Explanations for Natural Language Inference*
Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen and Yin Zhang

08:20–08:30 *Self-Guided Contrastive Learning for BERT Sentence Representations*
Taeuk Kim, Kang Min Yoo and Sang-goo Lee

08:30–08:40 *LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations*
Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu and Kai Yu

08:40–08:47 *DefSent: Sentence Embeddings using Definition Sentences*
Hayato Tsukagoshi, Ryohei Sasano and Koichi Takeda

08:47–08:54 *Discrete Cosine Transform as Universal Sentence Encoder*
Nada Almarwani and Mona Diab

Session 7C: Speech and Multimodality 1

08:00–08:10 *Multi-stage Pre-training over Simplified Multimodal Pre-training Models*
Tongtong Liu, Fangxiang Feng and Xiaojie WANG

08:10–08:20 *Beyond Sentence-Level End-to-End Speech Translation: Context Helps*
Biao Zhang, Ivan Titov, Barry Haddow and Rico Sennrich

08:20–08:30 *LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding*
Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang and Lidong Zhou

08:30–08:40 *UNIMO: Towards Unified-Modal Understanding and Generation via Cross-Modal Contrastive Learning*
Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu and Haifeng Wang

08:40–08:50 *Missing Modality Imagination Network for Emotion Recognition with Uncertain Missing Modalities*
Jinming Zhao, Ruichen Li and Qin Jin

08:50–09:00 *Stacked Acoustic-and-Textual Encoding: Integrating the Pre-trained Models into Speech Translation Encoders*
Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, shen huang, Qi Ju, Tong Xiao and Jingbo Zhu

Session 7D: Syntax: Tagging, Chunking, and Parsing 1

08:00–08:10 *N-ary Constituent Tree Parsing with Recursive Semi-Markov Model*
Xin Xin, Jinlong Li and Zeqi Tan

08:10–08:20 *Automated Concatenation of Embeddings for Structured Prediction*
Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang and Kewei Tu

08:20–08:30 *Multi-View Cross-Lingual Structured Prediction with Minimum Supervision*
Zechuan Hu, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang and Kewei Tu

08:30–08:40 *The Limitations of Limited Context for Constituency Parsing*
Yuchen Li and Andrej Risteski

Tuesday, August 3, 2021 (all times UTC+0) (continued)

08:40–08:50 *Neural Bi-Lexicalized PCFG Induction*
Songlin Yang, Yanpeng Zhao and Kewei Tu

Session 7E: Resources and Evaluation 2

08:00–08:10 *Ruddit: Norms of Offensiveness for English Reddit Comments*
Rishav Hada, Sohi Sudhir, Pushkar Mishra, Helen Yannakoudakis, Saif M. Mohammad and Ekaterina Shutova

08:10–08:20 *Towards Quantifiable Dialogue Coherence Evaluation*
Zheng Ye, Liucun Lu, Lishan Huang, Liang Lin and Xiaodan Liang

08:20–08:30 *Assessing the Representations of Idiomaticity in Vector Models with a Noun Compound Dataset Labeled at Type and Token Levels*
Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart and Aline Villavicencio

08:30–08:40 *Factoring Statutory Reasoning as Language Understanding Challenges*
Nils Holzenberger and Benjamin Van Durme

08:40–08:50 *Evaluating Evaluation Measures for Ordinal Classification and Ordinal Quantification*
Tetsuya Sakai

08:50–08:57 *AligNarr: Aligning Narratives on Movies*
Paramita Mirza, Mostafa Abouhamra and Gerhard Weikum

Session 8A: Information Extraction 4

09:00–09:10 *Interpretable and Low-Resource Entity Matching via Decoupling Feature Learning from Decision Making*
Zijun Yao, Chengjiang Li, Tiansi Dong, Xin Lv, Jifan Yu, Lei Hou, Juanzi Li, YICHI ZHANG and zelin Dai

09:10–09:20 *Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition*
Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang and Weiming Lu

09:20–09:30 *Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction*
Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao and Shaoyi Chen

09:30–09:40 *A Large-Scale Chinese Multimodal NER Dataset with Speech Clues*
Dianbo Sui, Zhengkun Tian, Yubo Chen, Kang Liu and Jun Zhao

09:40–09:50 *A Neural Transition-based Joint Model for Disease Named Entity Recognition and Normalization*
Zongcheng Ji, Tian Xia, Mei Han and Jing Xiao

09:50–10:00 *OntoED: Low-resource Event Detection with Ontology Embedding*
Shumin Deng, Ningyu Zhang, Luoqiu Li, Chen Hui, tou huaixiao, Mosha Chen, Fei Huang and Huajun Chen

Session 8B: Machine Translation and Multilinguality 4

09:00–09:10 *Self-Training Sampling with Monolingual Data Uncertainty for Neural Machine Translation*
Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu and Irwin King

09:10–09:20 *Breaking the Corpus Bottleneck for Context-Aware Neural Machine Translation with Cross-Task Pre-training*
Linqing Chen, Junhui Li, Zhengxian Gong, Boxing Chen, Weihua Luo, Min Zhang and Guodong Zhou

09:20–09:30 *Guiding Teacher Forcing with Seer Forcing for Neural Machine Translation*
Yang Feng, Shuhao Gu, Dengji Guo, Zhengxin Yang and Chenze Shao

09:30–09:40 *Cascade versus Direct Speech Translation: Do the Differences Still Make a Difference?*
Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri and Marco Turchi

Tuesday, August 3, 2021 (all times UTC+0) (continued)

09:40–09:50 *Unsupervised Neural Machine Translation for Low-Resource Domains via Meta-Learning*
Cheonbok Park, Yunwon Tae, TaeHee Kim, Soyoung Yang, Mohammad Azam Khan, Lucy Park and Jaegul Choo

09:50–09:57 *An Exploratory Analysis of Multilingual Word-Level Quality Estimation with Cross-Lingual Transformers*
Tharindu Ranasinghe, Constantin Orasan and Ruslan Mitkov

Session 8C: Machine Learning for NLP 4

09:00–09:10 *Lightweight Cross-Lingual Sentence Representation Learning*
Zhuoyuan Mao, Prakhar Gupta, Chenhui Chu, Martin Jaggi and Sadao Kurohashi

09:10–09:20 *ERNIE-Doc: A Retrospective Long-Document Modeling Transformer*
SiYu Ding, Junyuan Shang, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu and Haifeng Wang

09:20–09:30 *Marginal Utility Diminishes: Exploring the Minimum Knowledge for BERT Knowledge Distillation*
Yuanxin LIU, Fandong Meng, Zheng Lin, Weiping Wang and Jie Zhou

09:30–09:40 *Rational LAMOL: A Rationale-based Lifelong Learning Framework*
Kasidis Kanwatchara, Thanapapas Horsuwan, Piyawat Lertvittayakumjorn, Boonserm Kijsirikul and Peerapon Vateekul

09:40–09:50 *EnsLM: Ensemble Language Model for Data Diversity by Semantic Clustering*
Zhibin Duan, Hao Zhang, Chaojie Wang, Zhengjue Wang, Bo Chen and Mingyuan Zhou

09:50–10:00 *LeeBERT: Learned Early Exit for BERT with cross-level optimization*
Wei Zhu

Session 8D: NLP Applications 2

09:00–09:10 *Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering*
Reinald Adrian Pugoy and Hung-Yu Kao

09:10–09:20 *PLOME: Pre-training with Misspelled Knowledge for Chinese Spelling Correction*
Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang and Di Wang

09:20–09:30 *Competence-based Multimodal Curriculum Learning for Medical Report Generation*
Fenglin Liu, Shen Ge and Xian Wu

09:30–09:40 *Learning Syntactic Dense Embedding with Correlation Graph for Automatic Readability Assessment*
Xinying Qiu, Yuan Chen, Hanwu Chen, Jian-Yun Nie, Yuming Shen and Dawei Lu

09:40–09:50 *Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains*
Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Li and jun huang

09:50–09:57 *Exploration and Exploitation: Two Ways to Improve Chinese Spelling Correction Models*
Chong Li, Cenyuan Zhang, Xiaoqing Zheng and Xuanjing Huang

Session 8E: Question Answering 1

09:00–09:10 *A Semantic-based Method for Unsupervised Commonsense Question Answering*
Yilin Niu, Fei Huang, Jiaming Liang, Wenkai Chen, Xiaoyan Zhu and Minlie Huang

09:10–09:20 *Explanations for CommonsenseQA: New Dataset and Models*
Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla and Dinesh Garg

09:20–09:30 *Few-Shot Question Answering by Pretraining Span Selection*
Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson and Omer Levy

09:30–09:40 *UnitedQA: A Hybrid Approach for Open Domain Question Answering*
Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen and Jianfeng Gao

09:40–09:50 *Database reasoning over text*
James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel and Alon Halevy

09:50–09:57 *Training Adaptive Computation for Open-Domain Question Answering with Computational Constraints*
Yuxiang Wu, Pasquale Minervini, Pontus Stenetorp and Sebastian Riedel

Session 9A: Machine Translation and Multilinguality 5

10:00–10:10 *Online Learning Meets Machine Translation Evaluation: Finding the Best Systems with the Least Human Effort*
Vânia Mendonça, Ricardo Rei, Luisa Coheur, Alberto Sardinha and Ana Lúcia Santos

10:10–10:20 *How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models*
Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder and Iryna Gurevych

10:20–10:30 *Evaluating morphological typology in zero-shot cross-lingual transfer*
Antonio Martínez-García, Toni Badia and Jeremy Barnes

10:30–10:40 *From Machine Translation to Code-Switching: Generating High-Quality Code-Switched Text*
Ishan Tarunesh, Syamantak Kumar and Preethi Jyothi

10:40–10:50 *Fast and Accurate Neural Machine Translation with Translation Memory*
Qiuxiang He, Guoping Huang, Qu Cui, Li Li and Lemao Liu

10:50–10:57 *An Empirical Study on Adversarial Attack on NMT: Languages and Positions Matter*
Zhiyuan Zeng and Deyi Xiong

Session 9B: Resources and Evaluation 3

10:00–10:10 *Annotating Online Misogyny*
Philine Zeinert, Nanna Inie and Leon Derczynski

10:10–10:20 *Few-NERD: A Few-shot Named Entity Recognition Dataset*
Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng and Zhiyuan Liu

10:20–10:30 *MultiMET: A Multimodal Dataset for Metaphor Understanding*
Dongyu Zhang, Minghao Zhang, Heting Zhang, Liang Yang and Hongfei LIN

10:30–10:40 *Human-in-the-Loop for Data Collection: a Multi-Target Counter Narrative Dataset to Fight Online Hate Speech*
Margherita Fanton, Helena Bonaldi, Serra Sinem Tekiroğlu and Marco Guerini

10:40–10:47 *OntoGUM: Evaluating Contextualized SOTA Coreference Resolution on 12 More Genres*
Yilun Zhu, Sameer Pradhan and Amir Zeldes

Session 9C: Question Answering 2

10:00–10:10 *Can Generative Pre-trained Language Models Serve As Knowledge Bases for Closed-book QA?*
Cunxiang Wang, Pai Liu and Yue Zhang

10:10–10:20 *Joint Models for Answer Verification in Question Answering Systems*
Zeyu Zhang, Thuy Vu and Alessandro Moschitti

10:20–10:30 *Answering Ambiguous Questions through Generative Evidence Fusion and Round-Trip Prediction*
Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O. Arnold and Bing Xiang

10:30–10:40 *TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance*
Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng and Tat-Seng Chua

10:40–10:50 *Modeling Transitions of Focal Entities for Conversational Knowledge Base Question Answering*
Yunshi Lan and Jing Jiang

10:50–10:57 *In Factuality: Efficient Integration of Relevant Facts for Visual Question Answering*
Peter Vickers, Nikolaos Aletras, Emilio Monti and Loïc Barrault

Session 9D: Semantics: Sentence-level Semantics, Textual Inference and Other areas 3

10:00–10:10 *Evidence-based Factual Error Correction*
James Thorne and Andreas Vlachos

10:10–10:20 *Probabilistic, Structure-Aware Algorithms for Improved Variety, Accuracy, and Coverage of AMR Alignments*
Austin Blodgett and Nathan Schneider

10:20–10:30 *Meta-Learning to Compositionally Generalize*
Henry Conklin, Bailin Wang, Kenny Smith and Ivan Titov

10:30–10:40 *Taming Pre-trained Language Models with N-gram Representations for Low-Resource Domain Adaptation*
Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song and Tong Zhang

10:40–10:50 *ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning*
Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun and Jie Zhou

10:50–10:57 *Zero-shot Fact Verification by Claim Generation*
Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan and William Yang Wang

Session 9E: Sentiment Analysis, Stylistic Analysis, and Argument Mining 3

10:00–10:10 *Position Bias Mitigation: A Knowledge-Aware Graph Model for Emotion Cause Extraction*
Hanqi Yan, Lin Gui, Gabriele Pergola and Yulan He

10:10–10:20 *Every Bite Is an Experience: Key Point Analysis of Business Reviews*
Roy Bar-Haim, Lilach Eden, Yoav Kantor, Roni Friedman and Noam Slonim

10:20–10:30 *Structured Sentiment Analysis as Dependency Graph Parsing*
Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid and Erik Velldal

10:30–10:37 *Thank you BART! Rewarding Pre-Trained Models Improves Formality Style Transfer*
Huiyuan Lai, Antonio Toral and Malvina Nissim

10:37–10:44 *Deep Context- and Relation-Aware Learning for Aspect-based Sentiment Analysis*
Shinhyeok Oh, Dongyub Lee, Taesun Whang, IlNam Park, Seo Gaeun, EungGyun Kim and Harksoo Kim

10:44–10:51 *Towards Generative Aspect-Based Sentiment Analysis*
Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing and Wai Lam

Session 10A: Machine Translation and Multilinguality 6

11:00–11:10 *Consistency Regularization for Cross-Lingual Fine-Tuning*
Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song and Furu Wei

11:10–11:20 *Improving Pretrained Cross-Lingual Language Models via Self-Labeled Word Alignment*
Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang and Furu Wei

11:20–11:30 *Rejuvenating Low-Frequency Words: Making the Most of Parallel Data in Non-Autoregressive Translation*
Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao and Zhaopeng Tu

11:30–11:40 *G-Transformer for Document-Level Machine Translation*
Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen and Weihua Luo

Tuesday, August 3, 2021 (all times UTC+0) (continued)

11:40–11:50 *Prevent the Language Model from being Overconfident in Neural Machine Translation*
Mengqi Miao, Fandong Meng, Yijin Liu, Xiao-Hua Zhou and Jie Zhou

11:50–11:57 *Bilingual Mutual Information Based Adaptive Training for Neural Machine Translation*
Yangyifan Xu, Yijin Liu, Fandong Meng, Jiajun Zhang, Jinan Xu and Jie Zhou

Session 10B: Dialog and Interactive Systems 5

11:00–11:10 *Towards Emotional Support Dialog Systems*
Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang and Minlie Huang

11:10–11:20 *Novel Slot Detection: A Benchmark for Discovering Unknown Slot Types in the Task-Oriented Dialogue System*
Yanan Wu, Zhiyuan Zeng, Keqing He, Hong Xu, Yuanmeng Yan, Huixing Jiang and Weiran Xu

11:20–11:30 *GTM: A Generative Triple-wise Model for Conversational Question Generation*
Lei Shen, Fandong Meng, Jinchao Zhang, Yang Feng and Jie Zhou

11:30–11:40 *Diversifying Dialog Generation via Adaptive Label Smoothing*
Yida Wang, Yinhe Zheng, Yong Jiang and Minlie Huang

11:40–11:50 *Out-of-Scope Intent Detection with Self-Supervision and Discriminative Training*
Li-Ming Zhan, Haowen Liang, Bo LIU, Lu Fan, Xiao-Ming Wu and Albert Y.S. Lam

11:50–11:57 *Continual Learning for Task-oriented Dialogue System with Iterative Network Pruning, Expanding and Masking*
Binzong Geng, Fajie Yuan, Qiancheng Xu, Ying Shen, Ruifeng Xu and Min Yang

Session 10C: Information Extraction 5

11:00–11:10 *Document-level Event Extraction via Heterogeneous Graph-based Interaction Model with a Tracker*
Runxin Xu, Tianyu Liu, Lei Li and Baobao Chang

11:10–11:20 *Nested Named Entity Recognition via Explicitly Excluding the Influence of the Best Path*
Yiran Wang, Hiroyuki Shindo, Yuji Matsumoto and Taro Watanabe

11:20–11:30 *LearnDA: Learnable Knowledge-Guided Data Augmentation for Event Causality Identification*
Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng and Yuguang Chen

11:30–11:40 *Revisiting the Negative Data of Distantly Supervised Relation Extraction*
Chenhao Xie, Jiaqing Liang, Jingping Liu, Chengsong Huang, Wenhao Huang and Yanghua Xiao

11:40–11:50 *Knowing the No-match: Entity Alignment with Dangling Cases*
Zequn Sun, Muhao Chen and Wei Hu

11:50–11:57 *TIMERS: Document-level Temporal Relation Extraction*
Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran and Dinesh Manocha

Session 10D: Phonology, Morphology and Word Segmentation 1

11:00–11:10 *Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words*
Valentin Hofmann, Janet Pierrehumbert and Hinrich Schütze

11:10–11:20 *Optimizing over Subsequences Generates Context-Sensitive Languages*
Andrew Lamont

11:20–11:30 *Morphology Matters: A Multilingual Language Modeling Analysis*
Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu and Lane Schwartz

11:30–11:37 *Improving Arabic Diacritization with Regularized Decoding and Adversarial Training*
Han Qin, Guimin Chen, Yuanhe Tian and Yan Song

Tuesday, August 3, 2021 (all times UTC+0) (continued)

11:37–11:44 *When is Char Better Than Subword: A Systematic Study of Segmentation Algorithms for Neural Machine Translation*
Jiahuan Li, Yutong Shen, Shujian Huang, Xinyu Dai and Jiajun CHEN

11:44–11:51 *More than Text: Multi-modal Chinese Word Segmentation*
Dong Zhang, Zheng Hu, Shoushan Li, Hanqian Wu, Qiaoming Zhu and Guodong Zhou

Session 10E: Semantics: Lexical Semantics 1

11:00–11:10 *BERT is to NLP what AlexNet is to CV: Can Pre-Trained Language Models Identify Analogies?*
Asahi Ushio, Luis Espinosa Anke, Steven Schockaert and Jose Camacho-Collados

11:10–11:20 *Exploring the Representation of Word Meanings in Context: A Case Study on Homonymy and Synonymy*
Marcos Garcia

11:20–11:30 *Measuring Fine-Grained Domain Relevance of Terms: A Hierarchical Core-Fringe Approach*
Jie Huang, Kevin Chang, JinJun Xiong and Wen-mei Hwu

11:30–11:37 *A Mixture-of-Experts Model for Antonym-Synonym Discrimination*
Zhipeng Xie and Nan Zeng

11:37–11:44 *Learning Domain-Specialised Representations for Cross-Lingual Biomedical Entity Linking*
Fangyu Liu, Ivan Vulić, Anna Korhonen and Nigel Collier

11:44–11:51 *A Cluster-based Approach for Improving Isotropy in Contextual Embedding Space*
Sara Rajaee and Mohammad Taher Pilehvar

14:00–15:30 ***Business meeting and Green NLP panel***

15:30–16:30 ***Keynote 3. Christopher Potts: Reliable Characterizations of NLP Systems as a Social Responsibility***

Session 11A: Dialog and Interactive Systems 6

16:30–16:40 *HERALD: An Annotation Efficient Method to Detect User Disengagement in Social Conversations*
Weixin Liang, Kai-Hui Liang and Zhou Yu

16:40–16:50 *Value-Agnostic Conversational Semantic Parsing*
Emmanouil Antonios Platanios, Adam Pauls, Subhro Roy, Yuchen Zhang, Alexander Kyte, Alan Guo, Sam Thomson, Jayant Krishnamurthy, Jason Wolfe, Jacob Andreas and Dan Klein

16:50–17:00 *MPC-BERT: A Pre-Trained Language Model for Multi-Party Conversation Understanding*
Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng and Daxin Jiang

17:00–17:10 *Best of Both Worlds: Making High Accuracy Non-incremental Transformer-based Disfluency Detection Incremental*
Morteza Rohanian and Julian Hough

17:10–17:20 *NeuralWOZ: Learning to Collect Task-Oriented Dialogue via Model-Based Simulation*
Sungdong Kim, Minsuk Chang and Sang-Woo Lee

17:20–17:27 *Unsupervised Enrichment of Persona-grounded Dialog with Background Stories*
Bodhisattwa Prasad Majumder, Taylor Berg-Kirkpatrick, Julian McAuley and Harsh Jhamtani

Session 11B: Linguistic Theories, Cognitive Modeling and Psycholinguistics 1

16:30–16:40 *CDRNN: Discovering Complex Dynamics in Human Language Processing*
Cory Shain

16:40–16:50 *Structural Guidance for Transformer Language Models*
Peng Qian, Tahira Naseem, Roger Levy and Ramón Fernandez Astudillo

16:50–17:00 *Surprisal Estimators for Human Reading Times Need Character Models*
Byung-Doh Oh, Christian Clark and William Schuler

17:00–17:10 *CogAlign: Learning to Align Textual Neural Representations to Cognitive Language Processing Signals*
Yuqi Ren and Deyi Xiong

17:10–17:20 *Formal Basis of a Language Universal*
Milos Stanojevic and Mark Steedman

17:20–17:27 *Beyond Laurel/Yanny: An Autoencoder-Enabled Search for Polyperceivable Audio*
Kartik Chandra, Chuma Kabaghe and Gregory Valiant

Session 11C: Machine Learning for NLP 5

16:30–16:40 *Self-Attention Networks Can Process Bounded Hierarchical Languages*
Shunyu Yao, Binghui Peng, Christos Papadimitriou and Karthik Narasimhan

16:40–16:50 *TextSETTR: Few-Shot Text Style Extraction and Tunable Targeted Restyling*
Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David Uthus and Zarana Parekh

16:50–17:00 *H-Transformer-1D: Fast One-Dimensional Hierarchical Attention for Sequences*
Zhenhai Zhu and Radu Soricut

17:00–17:10 *Making Pre-trained Language Models Better Few-shot Learners*
Tianyu Gao, Adam Fisch and Danqi Chen

17:10–17:20 *A Sweet Rabbit Hole by DARCY: Using Honeypots to Detect Universal Trigger's Adversarial Attacks*
Thai Le, Noseong Park and Dongwon Lee

17:20–17:27 *Don't Let Discourse Confine Your Model: Sequence Perturbations for Improved Event Language Models*
Mahnaz Koupaee, Greg Durrett, Nathanael Chambers and Niranjan Balasubramanian

Session 11D: Information Retrieval and Text Mining 1

16:30–16:40 *Towards Propagation Uncertainty: Edge-enhanced Bayesian Graph Convolutional Networks for Rumor Detection*
Lingwei Wei, Dou Hu, Wei Zhou, Zhaojuan Yue and Songlin Hu

16:40–16:50 *Label-Specific Dual Graph Neural Network for Multi-Label Text Classification*
Qianwen Ma, Chunyuan Yuan, Wei Zhou and Songlin Hu

16:50–17:00 *TAN-NTM: Topic Attention Networks for Neural Topic Modeling*
Madhur Panwar, Shashank Shailabh, Milan Aggarwal and Balaji Krishnamurthy

17:00–17:10 *Cross-language Sentence Selection via Data Augmentation and Rationale Training*
Yanda Chen, Chris Kedzie, Suraj Nair, Petra Galuscakova, Rui Zhang, Douglas Oard and Kathleen McKeown

17:10–17:20 *A Neural Model for Joint Document and Snippet Ranking in Question Answering for Large Document Collections*
Dimitris Pappas and Ion Androutsopoulos

17:20–17:27 *The Curse of Dense Low-Dimensional Information Retrieval for Large Index Sizes*
Nils Reimers and Iryna Gurevych

Session 11E: Discourse and Pragmatics 1

16:30–16:40 *W-RST: Towards a Weighted RST-style Discourse Framework*
Patrick Huber, Wen Xiao and Giuseppe Carenini

16:40–16:50 *ABCD: A Graph Framework to Convert Complex Sentences to a Covering Set of Simple Sentences*
Yanjun Gao, Ting-Hao Huang and Rebecca J. Passonneau

16:50–17:00 *Which Linguist Invented the Lightbulb? Presupposition Verification for Question-Answering*
Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan and Deepak Ramachandran

17:00–17:10 *Adversarial Learning for Discourse Rhetorical Structure Parsing*
Longyin Zhang, Fang Kong and Guodong Zhou

Tuesday, August 3, 2021 (all times UTC+0) (continued)

17:10–17:20 *Exploring Discourse Structures for Argument Impact Classification*
Xin Liu, Jiefu Ou, Yangqiu Song and Xin Jiang

Session 12A: Machine Translation and Multilinguality 7

23:00–23:10 *Point, Disambiguate and Copy: Incorporating Bilingual Dictionaries for Neural Machine Translation*
Tong Zhang, Long Zhang, Wei Ye, Bo Li, Jinan Sun, Xiaoyu Zhu, Wen Zhao and Shikun Zhang

23:10–23:20 *VECO: Variable and Flexible Cross-lingual Pre-training for Language Understanding and Generation*
Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang and Luo Si

23:20–23:30 *A unified approach to sentence segmentation of punctuated text in many languages*
Rachel Wicks and Matt Post

23:30–23:40 *Towards User-Driven Neural Machine Translation*
Huan Lin, Liang Yao, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Degen Huang and Jinsong Su

23:40–23:50 *End-to-End Lexically Constrained Machine Translation for Morphologically Rich Languages*
Josef Jon, João Paulo Aires, Dusan Varis and Ondřej Bojar

23:50–23:57 *Cross-lingual Text Classification with Heterogeneous Graph Neural Network*
Ziyun Wang, Xuan Liu, Peiji Yang, Shixing Liu and zhisheng wang

Session 12B: Resources and Evaluation 4

23:00–23:10 *Handling Extreme Class Imbalance in Technical Logbook Datasets*
Farhad Akhbardeh, Cecilia Ovesdotter Alm, Marcos Zampieri and Travis Desell

23:10–23:20 *ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation*
Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya and Ashutosh Modi

23:20–23:30 *Supporting Cognitive and Emotional Empathic Writing of Students*
Thiemo Wambsganss, Christina Niklaus, Matthias Söllner, Siegfried Handschuh and Jan Marco Leimeister

23:30–23:40 *Context-aware Adversarial Training for Name Regularity Bias in Named Entity Recognition*
Abbas Ghaddar, Philippe Langlais, Ahmad Rashid and Mehdi Rezagholizadeh

23:40–23:50 *SummEval: Re-evaluating Summarization Evaluation*
Alex Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong and Richard Socher

23:50–24:00 *Towards Question-Answering as an Automatic Metric for Evaluating the Content Quality of a Summary*
Daniel Deutsch, Tania Bedrax-Weiss and Dan Roth

Session 12C: Question Answering 3

23:00–23:10 *Dual Reader-Parser on Hybrid Textual and Tabular Evidence for Open Domain Question Answering*
Alexander Hanbo Li, Patrick Ng, Peng Xu, Henghui Zhu, Zhiguo Wang and Bing Xiang

23:10–23:20 *Generation-Augmented Retrieval for Open-Domain Question Answering*
Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han and Weizhu Chen

23:20–23:30 *Check It Again:Progressive Visual Question Answering via Visual Entailment*
Qingyi Si, Zheng Lin, Ming yu Zheng, Peng Fu and Weiping Wang

23:30–23:40 *A Mutual Information Maximization Approach for the Spurious Solution Problem in Weakly Supervised Question Answering*
Zhihong Shao, Lifeng Shang, Qun Liu and Minlie Huang

23:40–23:50 *Relevance-guided Supervision for OpenQA with ColBERT*
Omar Khattab, Christopher Potts and Matei Zaharia

23:50–23:57 *Towards more equitable question answering systems: How much more data do you need?*
Arnab Debnath, Navid Rajabi, Fardina Fathmiul Alam and Antonios Anastasopoulos

Session 12D: Theme 1

23:00–23:10 *Breaking Down Walls of Text: How Can NLP Benefit Consumer Privacy?*
Abhilasha Ravichander, Alan W Black, Thomas Norton, Shomir Wilson and Norman Sadeh

23:10–23:20 *Supporting Land Reuse of Former Open Pit Mining Sites using Text Classification and Active Learning*
Christopher Schröder, Kim Bürgl, Yves Annanias, Andreas Niekler, Lydia Müller, Daniel Wiegreffe, Christian Bender, Christoph Mengs, Gerik Scheuermann and Gerhard Heyer

23:20–23:30 *Reliability Testing for Natural Language Processing Systems*
Samson Tan, Shafiq Joty, Kathy Baxter, Araz Taeihagh, Gregory A. Bennett and Min-Yen Kan

23:30–23:40 *Learning Language and Multimodal Privacy-Preserving Markers of Mood from Mobile Data*
Paul Pu Liang, Terrance Liu, Anna Cai, Michal Muszynski, Ryo Ishii, Nick Allen, Randy Auerbach, David Brent, Ruslan Salakhutdinov and Louis-Philippe Morency

23:40–23:50 *Anonymisation Models for Text Data: State of the art, Challenges and Future Directions*
Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet and Lilja Øvrelid

Wednesday, August 4, 2021 (all times UTC+0)

Poster 2A: Semantics: Sentence-level Semantics, Textual Inference and Other areas

0:00–2:00 *End-to-End AMR Corefencence Resolution*
Qiankun Fu, Linfeng Song, Wenyu Du and Yue Zhang

Poster 2B: Linguistic Theories, Cognitive Modeling and Psycholinguistics

0:00–2:00 *How is BERT surprised? Layerwise detection of linguistic anomalies*
Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu and Frank Rudzicz

0:00–2:00 *Psycholinguistic Tripartite Graph Network for Personality Detection*
Tao Yang, Feifan Yang, Haolan Ouyang and Xiaojun Quan

Poster 2C: Semantics: Lexical Semantics

0:00–2:00 *Verb Metaphor Detection via Contextual Relation Learning*
Wei Song, Shuhui Zhou, Ruiji Fu, Ting Liu and Lizhen Liu

Poster 2D: Speech and Multimodality

0:00–2:00 *Improving Speech Translation by Understanding and Learning from the Auxiliary Text Translation Task*
Yun Tang, Juan Pino, Xian Li, Changhan Wang and Dmitriy Genzel

Poster 2E: Ethics in NLP

0:00–2:00 *Probing Toxic Content in Large Pre-Trained Language Models*
Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song and Dit-Yan Yeung

0:00–2:00 *Societal Biases in Language Generation: Progress and Challenges*
Emily Sheng, Kai-Wei Chang, Prem Natarajan and Nanyun Peng

Poster 2F: Interpretability and Analysis of Models for NLP

0:00–2:00 *Reservoir Transformers*
Sheng Shen, Alexei Baevski, Ari Morcos, Kurt Keutzer, Michael Auli and Douwe Kiela

Poster 2G: Machine Learning for NLP

0:00–2:00 *Subsequence Based Deep Active Learning for Named Entity Recognition*
Puria Radmard, Yassir Fathullah and Aldo Lipani

0:00–2:00 *Convolutions and Self-Attention: Re-interpreting Relative Positions in Pre-trained Language Models*
Tyler Chang, Yifan Xu, Weijian Xu and Zhuowen Tu

0:00–2:00 *BinaryBERT: Pushing the Limit of BERT Quantization*
Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin JIN, Xin Jiang, Qun Liu, Michael Lyu and Irwin King

0:00–2:00 *Embedding Time Differences in Context-sensitive Neural Networks for Learning Time to Event*
Nazanin Dehghani, Hassan Hajipoor and Hadi Amiri

0:00–2:00 *Are Pretrained Convolutions Better than Pretrained Transformers?*
Yi Tay, Mostafa Dehghani, Jai Prakash Gupta, Vamsi Aribandi, Dara Bahri, Zhen Qin and Donald Metzler

0:00–2:00 *PairRE: Knowledge Graph Embeddings via Paired Relation Vectors*
Linlin Chao, Jianshan He, Taifeng Wang and Wei Chu

0:00–2:00 *Improving Compositional Generalization in Classification Tasks via Structure Annotations*
Juyong Kim, Pradeep Ravikumar, Joshua Ainslie and Santiago Ontanon

0:00–2:00 *Learning to Generate Task-Specific Adapters from Task Description*
Qinyuan Ye and Xiang Ren

0:00–2:00 *Hierarchy-aware Label Semantics Matching Network for Hierarchical Text Classification*
Haibin Chen, Qianli Ma, Zhenxi Lin and Jiangyue Yan

0:00–2:00 *HiddenCut: Simple Data Augmentation for Natural Language Understanding with Better Generalizability*
Jiaao Chen, Dinghan Shen, Weizhu Chen and Diyi Yang

0:00–2:00 *Efficient Content-Based Sparse Attention with Routing Transformers*
Aurko Roy, Mohammad Saffar, Ashish Vaswani and David Grangier

Poster 2H: Dialog and Interactive Systems

0:00–2:00 *Neural Stylistic Response Generation with Disentangled Latent Variables*
Qingfu Zhu, Wei-Nan Zhang, Ting Liu and William Yang Wang

0:00–2:00 *Intent Classification and Slot Filling for Privacy Policies*
Wasi Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian and Kai-Wei Chang

0:00–2:00 *RADDLE: An Evaluation Benchmark and Analysis Platform for Robust Task-oriented Dialog Systems*
Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li and Jianfeng Gao

0:00–2:00 *QA-Driven Zero-shot Slot Filling with Weak Supervision Pretraining*
Xinya Du, Luheng He, Qi Li, Dian Yu, Panupong Pasupat and Yuan Zhang

0:00–2:00 *Domain-Adaptive Pretraining Methods for Dialogue Understanding*
Han Wu, Kun Xu, Linfeng Song, Lifeng Jin, Haisong Zhang and Linqi Song

0:00–2:00 *Semantic Representation for Dialogue Modeling*
Xuefeng Bai, Yulong Chen, Linfeng Song and Yue Zhang

0:00–2:00 *A Pre-training Strategy for Zero-Resource Response Selection in Knowledge-Grounded Conversations*
Chongyang Tao, Changyu Chen, Jiazhan Feng, Ji-Rong Wen and Rui Yan

0:00–2:00 *SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching*
Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden and Jianfeng Gao

Poster 2I: Information Retrieval and Text Mining

0:00–2:00 *Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks*
Yuanhe Tian, Guimin Chen, Yan Song and Xiang Wan

0:00–2:00 *Evaluating Entity Disambiguation and the Role of Popularity in Retrieval-Based NLP*
Anthony Chen, Pallavi Gudipati, Shayne Longpre, Xiao Ling and Sameer Singh

Poster 2J: Resources and Evaluation

0:00–2:00 *Targeting the Benchmark: On Methodology in Current Natural Language Processing Research*
David Schlangen

0:00–2:00 *Evaluation Examples are not Equally Informative: How should that change NLP Leaderboards?*
Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia and Jordan Boyd-Graber

Poster 2K: Computational Social Science and Cultural Analytics

0:00–2:00 *Claim Matching Beyond English to Scale Global Fact-Checking*
Ashkan Kazemi, Kiran Garimella, Devin Gaffney and Scott Hale

0:00–2:00 *X-Fact: A New Benchmark Dataset for Multilingual Fact Checking*
Ashim Gupta and Vivek Srikumar

Poster 2L: Machine Translation and Multilinguality

0:00–2:00 *SemFace: Pre-training Encoder and Decoder with a Semantic Interface for Neural Machine Translation*
Shuo Ren, Long Zhou, Shujie Liu, Furu Wei, Ming Zhou and Shuai Ma

0:00–2:00 *Energy-Based Reranking: Improving Neural Machine Translation Using Energy-Based Models*
Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer and Andrew McCallum

0:00–2:00 *nmT5 - Is parallel data still relevant for pre-training massively multilingual language models?*
Mihir Kale, Aditya Siddhant, Rami Al-Rfou, Linting Xue, Noah Constant and Melvin Johnson

0:00–2:00 *Syntax-augmented Multilingual BERT for Cross-lingual Transfer*
Wasi Ahmad, Haoran Li, Kai-Wei Chang and Yashar Mehdad

0:00–2:00 *How to Adapt Your Pretrained Multilingual Model to 1600 Languages*
Abteen Ebrahimi and Katharina Kann

0:00–2:00 *Synthesizing Parallel Data of User-Generated Texts with Zero-Shot Neural Machine Translation*
Benjamin Marie and Atsushi Fujita

Poster 2M: Syntax: Tagging, Chunking, and Parsing

0:00–2:00 *Weakly Supervised Named Entity Tagging with Learnable Logical Rules*
Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley and Zhe Feng

Poster 2N: NLP Applications

0:00–2:00 *Question Generation for Adaptive Education*
Megha Srivastava and Noah Goodman

Poster 2O: Language Generation

0:00–2:00 *Prefix-Tuning: Optimizing Continuous Prompts for Generation*
Xiang Lisa Li and Percy Liang

0:00–2:00 *One2Set: Generating Diverse Keyphrases as a Set*
Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu and Qi Zhang

0:00–2:00 *A Simple Recipe for Multilingual Grammatical Error Correction*
Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause and Aliaksei Severyn

0:00–2:00 *Continuous Language Generative Flow*
Zineng Tang, Shiyue Zhang, Hyounghun Kim and Mohit Bansal

0:00–2:00 *RYANSQL: Recursively Applying Sketch-based Slot Fillings for Complex Text-to-SQL in Cross-Domain Databases*
DongHyun Choi, Myeong Cheol Shin, EungGyun Kim and Dong Ryeol Shin

Poster 2P: Summarization

0:00–2:00 *TWAG: A Topic-Guided Wikipedia Abstract Generator*
Fangwei Zhu, Shangqing Tu, Jiaxin Shi, Juanzi Li, Lei Hou and Tong Cui

Poster 2Q: Question Answering

0:00–2:00 *Towards Visual Question Answering on Pathology Images*
Xuehai He, Zhuo Cai, Wenlan Wei, Yichen Zhang, Luntian Mou, Eric Xing and Pengtao Xie

0:00–2:00 *ForecastQA: A Question Answering Challenge for Event Forecasting with Temporal Text Data*
Woojeong Jin, Rahul Khanna, Suji Kim, Dong-Ho Lee, Fred Morstatter, Aram Galstyan and Xiang Ren

0:00–2:00 *Recursive Tree-Structured Self-Attention for Answer Sentence Selection*
Khalil Mrini, Emilia Farcas and Ndapa Nakashole

Poster 2R: Language Grounding to Vision, Robotics and Beyond

0:00–2:00 *Efficient Text-based Reinforcement Learning by Jointly Leveraging State and Commonsense Graph Representations*
Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Kartik Talamadupula, Mrinmaya Sachan and Murray Campbell

0:00–2:00 *mTVR: Multilingual Moment Retrieval in Videos*
Jie Lei, Tamara Berg and Mohit Bansal

Poster 2S: Information Extraction

0:00–2:00 *How Knowledge Graph and Attention Help? A Qualitative Analysis into Bag-level Relation Extraction*
Zikun Hu, Yixin Cao, Lifu Huang and Tat-Seng Chua

0:00–2:00 *Trigger is Not Sufficient: Exploiting Frame-aware Knowledge for Implicit Event Argument Extraction*
Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi and li jin

0:00–2:00 *Element Intervention for Open Relation Extraction*
Fangchao Liu, Lingyong Yan, Hongyu Lin, Xianpei Han and Le Sun

0:00–2:00 *Explicitly Capturing Relations between Entity Mentions via Graph Neural Networks for Domain-specific Named Entity Recognition*
Pei Chen, Haibo Ding, Jun Araki and Ruihong Huang

0:00–2:00 *AdaTag: Multi-Attribute Value Extraction from Product Profiles with Adaptive Decoding*
Jun Yan, Nasser Zalmout, Yan Liang, Christan Grant, Xiang Ren and Xin Luna Dong

0:00–2:00 *CoRI: Collective Relation Integration with Data Augmentation for Open Information Extraction*
Zhengbao Jiang, Jialong Han, BUNYAMIN SISMAN and Xin Luna Dong

0:00–2:00 *Benchmarking Scalable Methods for Streaming Cross Document Entity Coreference*
Robert L Logan IV, Andrew McCallum, Sameer Singh and Dan Bikel

0:00–2:00 *Search from History and Reason for Future: Two-stage Reasoning on Temporal Knowledge Graphs*
Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang and Xueqi Cheng

Poster 2T: Sentiment Analysis, Stylistic Analysis, and Argument Mining

0:00–2:00 *Employing Argumentation Knowledge Graphs for Neural Argument Generation*
Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou and Benno Stein

0:00–2:00 *Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction*
Lu Xu, Yew Ken Chia and Lidong Bing

Session 13A: Machine Translation and Multilinguality 8

08:00–08:10 *On Compositional Generalization of Neural Machine Translation*
Yafu Li, Yongjing Yin, Yulong Chen and Yue Zhang

08:10–08:20 *Mask-Align: Self-Supervised Neural Word Alignment*
Chi Chen, Maosong Sun and Yang Liu

08:20–08:30 *GWLAN: General Word-Level AutocompletioN for Computer-Aided Translation*
Huayang Li, Lemao Liu, Guoping Huang and Shuming Shi

08:30–08:37 *Improving Lexically Constrained Neural Machine Translation with Source-Conditioned Masked Span Prediction*
Gyubok Lee, Seongjun Yang and Edward Choi

Session 13B: Information Extraction 6

08:00–08:10 *De-biasing Distantly Supervised Named Entity Recognition via Causal Intervention*
Wenkai Zhang, Hongyu Lin, Xianpei Han and Le Sun

08:10–08:20 *A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition*
Fei Li, ZhiChao Lin, Meishan Zhang and Donghong Ji

08:20–08:30 *MLBiNet: A Cross-Sentence Collective Event Detection Network*
Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang and Huajun Chen

08:30–08:40 *Exploiting Document Structures and Cluster Consistencies for Event Coreference Resolution*
Hieu Minh Tran, Duy Phung and Thien Huu Nguyen

08:40–08:50 *StereoRel: Relational Triple Extraction from a Stereoscopic Perspective*
Xuetao Tian, Liping Jing, Lu He and Feng Liu

08:50–09:00 *Knowledge-Enriched Event Causality Identification via Latent Structure Induction Networks*
Pengfei Cao, Xinyu Zuo, Yubo Chen, Kang Liu, Jun Zhao, Yuguang Chen and Weihua Peng

Session 13C: Machine Learning for NLP 6

08:00–08:10 *Turn the Combination Lock: Learnable Textual Backdoor Attacks via Word Substitution*
Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu and Maosong Sun

08:10–08:20 *Parameter-Efficient Transfer Learning with Diff Pruning*
Demi Guo, Alexander Rush and Yoon Kim

08:20–08:30 *R2D2: Recursive Transformer based on Differentiable Tree for Interpretable Hierarchical Language Modeling*
Xiang Hu, Haitao Mi, Zujie Wen, Yafang Wang, Yi Su, Jing Zheng and Gerard de Melo

08:30–08:40 *Risk Minimization for Zero-shot Sequence Labeling*
Zechuan Hu, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang and Kewei Tu

08:40–08:50 *WARP: Word-level Adversarial ReProgramming*
Karen Hambardzumyan, Hrant Khachatrian and Jonathan May

08:50–09:00 *Lexicon Learning for Few Shot Sequence Modeling*
Ekin Akyurek and Jacob Andreas

Session 13D: NLP Applications 3

08:00–08:10 *Personalized Transformer for Explainable Recommendation*
Lei Li, Yongfeng Zhang and Li Chen

08:10–08:20 *Generating SOAP Notes from Doctor-Patient Conversations Using Modular Summarization Techniques*
Kundan Krishna, Sopan Khosla, Jeffrey Bigham and Zachary C. Lipton

08:20–08:30 *Tail-to-Tail Non-Autoregressive Sequence Prediction for Chinese Grammatical Error Correction*
Piji Li and Shuming Shi

08:30–08:40 *Early Detection of Sexual Predators in Chats*
Matthias Vogt, Ulf Leser and Alan Akbik

08:40–08:50 *Writing by Memorizing: Hierarchical Retrieval-based Medical Report Generation*
Xingyi Yang, Muchao Ye, Quanzeng You and Fenglong Ma

08:50–08:57 *Quotation Recommendation and Interpretation Based on Transformation from Queries to Quotations*
Lingzhi Wang, Xingshan Zeng and Kam-Fai Wong

Session 13E: Information Retrieval and Text Mining 2

08:00–08:10 *Concept-Based Label Embedding via Dynamic Routing for Hierarchical Text Classification*
Xuepeng Wang, Li Zhao, Bing Liu, Tao Chen, Feng Zhang and Di Wang

08:10–08:20 *VisualSparta: An Embarrassingly Simple Approach to Large-scale Text-to-Image Search with Weighted Bag-of-words*
Xiaopeng Lu, Tiancheng Zhao and Kyusong Lee

08:20–08:30 *Few-Shot Text Ranking with Meta Adapted Synthetic Weak Supervision*
Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu and Paul Bennett

08:30–08:40 *Semi-Supervised Text Classification with Balanced Deep Representation Distributions*
Changchun Li, Ximing Li and Jihong Ouyang

08:40–08:50 *Improving Document Representations by Generating Pseudo Query Embeddings for Dense Retrieval*
Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang and Wei Wu

08:50–08:57 *Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence*
Federico Bianchi, Silvia Terragni and Dirk Hovy

Poster 3A: Semantics: Sentence-level Semantics, Textual Inference and Other areas

9:00–11:00 *ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer*
Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu and Weiran Xu

9:00–11:00 *Exploring Dynamic Selection of Branch Expansion Orders for Code Generation*
Hui Jiang, Chulun Zhou, Fandong Meng, Biao Zhang, Jie Zhou, Degen Huang, Qingqiang Wu and Jinsong Su

9:00–11:00 *COINS: Dynamically Generating COntextualized Inference Rules for Narrative Story Completion*
Debjit Paul and Anette Frank

9:00–11:00 *Reasoning over Entity-Action-Location Graph for Procedural Text Understanding*
Hao Huang, Xiubo Geng, Jian Pei, Guodong Long and Daxin Jiang

Wednesday, August 4, 2021 (all times UTC+0) (continued)

9:00–11:00 *From Paraphrasing to Semantic Parsing: Unsupervised Semantic Parsing via Synchronous Semantic Decoding*
Shan Wu, Bo Chen, Chunlei Xin, Xianpei Han, Le Sun, Weipeng Zhang, Jiansong Chen, Fan Yang and Xunliang Cai

9:00–11:00 *Pre-training Universal Language Representation*
Yian Li and Hai Zhao

9:00–11:00 *Structural Pre-training for Dialogue Comprehension*
Zhuosheng Zhang and Hai Zhao

9:00–11:00 *AutoTinyBERT: Automatic Hyper-parameter Optimization for Efficient Pre-trained Language Models*
Yichun Yin, Cheng Chen, Lifeng Shang, Xin Jiang, Xiao Chen and Qun Liu

9:00–11:00 *Data Augmentation with Adversarial Training for Cross-Lingual NLI*
Xin Dong, Yaxin Zhu, Zuohui Fu, Dongkuan Xu and Gerard de Melo

9:00–11:00 *Input Representations for Parsing Discourse Representation Structures: Comparing English with Chinese*
Chunliu Wang, Rik van Noord, Arianna Bisazza and Johan Bos

9:00–11:00 *Code Generation from Natural Language with Less Prior Knowledge and More Monolingual Data*
Sajad Norouzi, Keyi Tang and Yanshuai Cao

9:00–11:00 *Bootstrapped Unsupervised Sentence Representation Learning*
Yan Zhang, Ruidan He, ZUOZHU LIU, Lidong Bing and Haizhou Li

9:00–11:00 *Learning Event Graph Knowledge for Abductive Reasoning*
Li Du, Xiao Ding, Ting Liu and Bing Qin

9:00–11:00 *Issues with Entailment-based Zero-shot Text Classification*
Tingting Ma, Jin-Ge Yao, Chin-Yew Lin and Tiejun Zhao

9:00–11:00 *Neural-Symbolic Commonsense Reasoner with Relation Predictors*
Farhad Moghimifar, Lizhen Qu, Terry Yue Zhuo, Gholamreza Haffari and Mahsa Baktashmotlagh

Poster 3B: Linguistic Theories, Cognitive Modeling and Psycholinguistics

9:00–11:00 *A Cognitive Regularizer for Language Modeling*
Jason Wei, Clara Meister and Ryan Cotterell

9:00–11:00 *What Motivates You? Benchmarking Automatic Detection of Basic Needs from Short Posts*
Sanja Stajner, Seren Yenikent, Bilal Ghanem and Marc Franco-Salvador

9:00–11:00 *Lower Perplexity is Not Always Human-Like*
Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara and Kentaro Inui

Poster 3C: Semantics: Lexical Semantics

9:00–11:00 *Word Sense Disambiguation: Towards Interactive Context Exploitation from Both Word and Sense Perspectives*
Ming Wang and Yinglin Wang

9:00–11:00 *A Knowledge-Guided Framework for Frame Identification*
Xuefeng Su, Ru Li, Xiaoli Li, Jeff Z. Pan, Hu Zhang, Qinghua Chai and Xiaoqi Han

9:00–11:00 *Obtaining Better Static Word Embeddings Using Contextual Embedding Models*
Prakhar Gupta and Martin Jaggi

9:00–11:00 *Meta-Learning with Variational Semantic Memory for Word Sense Disambiguation*
Yingjun Du, Nithin Holla, Xiantong Zhen, Cees Snoek and Ekaterina Shutova

9:00–11:00 *LexFit: Lexical Fine-Tuning of Pretrained Language Models*
Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen and Goran Glavaš

9:00–11:00 *Semantic Frame Induction using Masked Word Embeddings and Two-Step Clustering*
Kosuke Yamada, Ryohei Sasano and Koichi Takeda

9:00–11:00 *Multi-SimLex: A Large-Scale Evaluation of Multilingual and Cross-Lingual Lexical Semantic Similarity*
Ivan Vulic, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, Roi Reichart and Anna Korhonen

Poster 3D: Speech and Multimodality

9:00–11:00 *Text-Free Image-to-Speech Synthesis Using Learned Segmental Units*
Wei-Ning Hsu, David Harwath, Tyler Miller, Christopher Song and James Glass

9:00–11:00 *CTFN: Hierarchical Learning for Multimodal Sentiment Analysis Using Coupled-Translation Fusion Network*
Jiajia Tang, Kang Li, Xuanyu Jin, Andrzej Cichocki, Qibin Zhao and Wanzeng Kong

9:00–11:00 *Lightweight Adapter Tuning for Multilingual Speech Translation*
Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab and Laurent Besacier

Poster 3E: Interpretability and Analysis of Models for NLP

9:00–11:00 *Parameter Selection: Why We Should Pay More Attention to It*
Jie-Jyun Liu, Tsung-Han Yang, Si-An Chen and Chih-Jen Lin

9:00–11:00 *Positional Artefacts Propagate Through Masked Language Model Embeddings*
Ziyang Luo, Artur Kulmizev and Xiaoxi Mao

9:00–11:00 *Language Model Evaluation Beyond Perplexity*
Clara Meister and Ryan Cotterell

9:00–11:00 *Learning to Explain: Generating Stable Explanations Fast*
Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf and Gholamreza Haffari

9:00–11:00 *StereoSet: Measuring stereotypical bias in pretrained language models*
Moin Nadeem, Anna Bethke and Siva Reddy

9:00–11:00 *Alignment Rationale for Natural Language Inference*
Zhongtao Jiang, Yuanzhe Zhang, Zhao Yang, Jun Zhao and Kang Liu

9:00–11:00 *Enabling Lightweight Fine-tuning for Pre-trained Language Model Compression based on Matrix Product Operators*
Peiyu Liu, Ze-Feng Gao, Wayne Xin Zhao, Zhi-Yuan Xie, Zhong-Yi Lu and Ji-Rong Wen

9:00–11:00 *On Sample Based Explanation Methods for NLP: Faithfulness, Efficiency and Semantic Evaluation*
Wei Zhang, Ziming Huang, Yada Zhu, Guangnan Ye, Xiaodong Cui and Fan Zhang

9:00–11:00 *CausaLM: Causal Model Explanation Through Counterfactual Language Models*
Amir Feder, Nadav Oved, Uri Shalit and Roi Reichart

9:00–11:00 *Amnesic Probing: Behavioral Explanation With Amnesic Counterfactuals*
Yanai Elazar, Shauli Ravfogel, Alon Jacovi and Yoav Goldberg

Poster 3F: Information Retrieval and Text Mining

9:00–11:00 *Syntax-Enhanced Pre-trained Model*
Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang and Nan Duan

9:00–11:00 *Matching Distributions between Model and Data: Cross-domain Knowledge Distillation for Unsupervised Domain Adaptation*
Bo Zhang, Xiaoming Zhang, Yun Liu, Lei Cheng and Zhoujun Li

9:00–11:00 *Counterfactual Inference for Text Classification Debiasing*
Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma and Pengjun Xie

9:00–11:00 *HieRec: Hierarchical User Interest Modeling for Personalized News Recommendation*
Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie and Yongfeng Huang

9:00–11:00 *Distinct Label Representations for Few-Shot Text Classification*
Sora Ohashi, Junya Takayama, Tomoyuki Kajiwara and Yuki Arase

9:00–11:00 *PP-Rec: News Recommendation with Personalized User Interest and Time-aware News Popularity*
Tao Qi, Fangzhao Wu, Chuhan Wu and Yongfeng Huang

9:00–11:00 *Article Reranking by Memory-Enhanced Key Sentence Matching for Detecting Previously Fact-Checked Claims*
Qiang Sheng, Juan Cao, Xueyao Zhang, Xirong Li and Lei Zhong

9:00–11:00 *Learning to Solve NLP Tasks in an Incremental Number of Languages*
Giuseppe Castellucci, Simone Filice, Danilo Croce and Roberto Basili

Poster 3G: Machine Learning for NLP

9:00–11:00 *Defense against Synonym Substitution-based Adversarial Attacks via Dirichlet Neighborhood Ensemble*
Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang and Xuanjing Huang

9:00–11:00 *Shortformer: Better Language Modeling using Shorter Inputs*
Ofir Press, Noah A. Smith and Mike Lewis

9:00–11:00 *BanditMTL: Bandit-based Multi-task Learning for Text Classification*
Yuren Mao, Zekai Wang, Weiwei Liu, Xuemin Lin and Wenbin Hu

9:00–11:00 *Unified Interpretation of Softmax Cross-Entropy and Negative Sampling: With Case Study for Knowledge Graph Embedding*
Hidetaka Kamigaito and Katsuhiko Hayashi

9:00–11:00 *Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling*
Chuhan Wu, Fangzhao Wu, Tao Qi and Yongfeng Huang

9:00–11:00 *De-Confounded Variational Encoder-Decoder for Logical Table-to-Text Generation*
Wenqing Chen, Jidong Tian, Yitian Li, Hao He and Yaohui Jin

9:00–11:00 *Rethinking Stealthiness of Backdoor Attack against NLP Models*
Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou and Xu Sun

9:00–11:00 *Crowdsourcing Learning as Domain Adaptation: A Case Study on Named Entity Recognition*
Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang and Pengjun Xie

9:00–11:00 *Robust Transfer Learning with Pretrained Language Models through Adapters*
Wenjuan Han, Bo Pang and Ying Nian Wu

9:00–11:00 *Embracing Ambiguity: Shifting the Training Target of NLI Models*
Johannes Mario Meissner, Napat Thumwanit, Saku Sugawara and Akiko Aizawa

9:00–11:00 *Exploring Distantly-Labeled Rationales in Neural Network Models*
Quzhe Huang, Shengqi Zhu, Yansong Feng and Dongyan Zhao

9:00–11:00 *Learning to Perturb Word Embeddings for Out-of-distribution QA*
Seanie Lee, Minki Kang, Juho Lee and Sung Ju Hwang

Poster 3H: Dialog and Interactive Systems

9:00–11:00 *Maria: A Visual Experience Powered Conversational Agent*
Zujie Liang, Huang Hu, Can Xu, Chongyang Tao, Xiubo Geng, yining Chen, Fan Liang and Daxin Jiang

9:00–11:00 *A Human-machine Collaborative Framework for Evaluating Malevolence in Dialogues*
Yangjun Zhang, Pengjie Ren and Maarten de Rijke

9:00–11:00 *Generating Relevant and Coherent Dialogue Responses using Self-Separated Conditional Variational AutoEncoders*
Bin Sun, Shaoxiong Feng, Yiwei Li, Jiamou Liu and Kan Li

9:00–11:00 *Modeling Discriminative Representations for Out-of-Domain Detection with Supervised Contrastive Learning*
Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang and Weiran Xu

9:00–11:00 *Learning to Ask Conversational Questions by Optimizing Levenshtein Distance*
Zhongkun Liu, Pengjie Ren, Zhumin CHEN, Zhaochun Ren, Maarten de Rijke and Ming Zhou

9:00–11:00 *DVD: A Diagnostic Dataset for Multi-step Reasoning in Video Grounded Dialogue*
Hung Le, Chinnadhurai Sankar, Seungwhan Moon, Ahmad Beirami, Alborz Geramifard and Satwik Kottur

9:00–11:00 *Preview, Attend and Review: Schema-Aware Curriculum Learning for Multi-Domain Dialogue State Tracking*
Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si and Xiaodan Zhu

9:00–11:00 *On the Generation of Medical Dialogs for COVID-19*
Meng Zhou, Zechen Li, Bowen Tan, Guangtao Zeng, Wenmian Yang, Xuehai He, Zeqian Ju, Subrato Chakravorty, Shu Chen, Xingyi Yang, Yichen Zhang, Qingyang Wu, Zhou Yu, Kun Xu, Eric Xing and Pengtao Xie

9:00–11:00 *Constructing Multi-Modal Dialogue Dataset by Replacing Text with Semantically Relevant Images*
Nyoungwoo Lee, Suwon Shin, Jaegul Choo, Ho-Jin Choi and Sung-Hyon Myaeng

9:00–11:00 *MMGCN: Multimodal Fusion via Deep Graph Convolution Network for Emotion Recognition in Conversation*
Jingwen Hu, Yuchen Liu, Jinming Zhao and Qin Jin

9:00–11:00 *DynaEval: Unifying Turn and Dialogue Level Evaluation*
Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee and Haizhou Li

9:00–11:00 *Unsupervised Learning of KB Queries in Task-Oriented Dialogs*
Dinesh Raghu, Nikhil Gupta and Mausam

Poster 3I: Ethics in NLP

9:00–11:00 *Exposing the limits of Zero-shot Cross-lingual Hate Speech Detection*
Debora Nozza

Poster 3J: Resources and Evaluation

9:00–11:00 *CoSQA: 20,000+ Web Queries for Code Search and Question Answering*
Junjie Huang, Duyu Tang, Linjun Shou, Ming Gong, Ke Xu, Daxin Jiang, Ming Zhou and Nan Duan

9:00–11:00 *QED: A Framework and Dataset for Explanations in Question Answering*
Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares and Michael Collins

Poster 3K: Machine Translation and Multilinguality

9:00–11:00 *Rewriter-Evaluator Architecture for Neural Machine Translation*
Yangming Li and Kaisheng Yao

9:00–11:00 *BERTTune: Fine-Tuning Neural Machine Translation with BERTScore*
Inigo Jauregi Unanue, Jacob Parnell and Massimo Piccardi

9:00–11:00 *Modeling Bilingual Conversational Characteristics for Neural Chat Translation*
Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu and Jie Zhou

9:00–11:00 *Importance-based Neuron Allocation for Multilingual Neural Machine Translation*
Wanying Xie, Yang Feng, Shuhao Gu and Dong Yu

9:00–11:00 *Transfer Learning for Sequence Generation: from Single-source to Multi-source*
Xuancheng Huang, jingfang xu, Maosong Sun and Yang Liu

9:00–11:00 *A Closer Look at Few-Shot Crosslingual Transfer: The Choice of Shots Matters*
Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen and Hinrich Schütze

Poster 3L: Discourse and Pragmatics

9:00–11:00 *Coreference Reasoning in Machine Reading Comprehension*
Mingzhu Wu, Nafise Sadat Moosavi, Dan Roth and Iryna Gurevych

9:00–11:00 *Entity Enhancement for Implicit Discourse Relation Classification in the Biomedical Domain*
Wei Shi and Vera Demberg

9:00–11:00 *Adapting Unsupervised Syntactic Parsing Methodology for Discourse Dependency Parsing*
Liwen Zhang, Ge Wang, Wenjuan Han and Kewei Tu

9:00–11:00 *Unsupervised Pronoun Resolution via Masked Noun-Phrase Prediction*
Ming Shen, Pratyay Banerjee and Chitta Baral

Poster 3M: Syntax: Tagging, Chunking, and Parsing

9:00–11:00 *A Conditional Splitting Framework for Efficient Constituency Parsing*
Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty and Xiaoli Li

9:00–11:00 *A Unified Generative Framework for Various NER Subtasks*
Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang and Xipeng Qiu

9:00–11:00 *An In-depth Study on Internal Structure of Chinese Words*
Chen Gong, Saihao Huang, Houquan Zhou, Zhenghua Li, Min Zhang, Zhefeng Wang, baoxing Huai and Nicholas Jing Yuan

9:00–11:00 *MulDA: A Multilingual Data Augmentation Framework for Low-Resource Cross-Lingual NER*
Linlin Liu, BOSHENG DING, Lidong Bing, Shafiq Joty, Luo Si and Chunyan Miao

9:00–11:00 *Lexicon Enhanced Chinese Sequence Labeling Using BERT Adapter*
Wei Liu, Xiyan Fu, Yue Zhang and Wenming Xiao

Poster 3N: NLP Applications

9:00–11:00 *Math Word Problem Solving with Explicit Numerical Values*
Qinzhuo Wu, Qi Zhang, Zhongyu Wei and Xuanjing Huang

9:00–11:00 *Neural-Symbolic Solver for Math Word Problems with Auxiliary Tasks*
Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang and Liang Lin

9:00–11:00 *SMedBERT: A Knowledge-Enhanced Pre-trained Language Model with Structured Semantics for Medical Text Mining*
Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang and XIAOFENG HE

9:00–11:00 *What is Your Article Based On? Inferring Fine-grained Provenance*
Yi Zhang, Zachary Ives and Dan Roth

9:00–11:00 *Cross-modal Memory Networks for Radiology Report Generation*
Zhihong Chen, Yaling Shen, Yan Song and Xiang Wan

9:00–11:00 *Controversy and Conformity: from Generalized to Personalized Aggressiveness Detection*
Kamil Kanclerz, Alicja Figas, Marcin Gruza, Tomasz Kajdanowicz, Jan Kocon, Daria Puchalska and Przemyslaw Kazienko

9:00–11:00 *Multi-perspective Coherent Reasoning for Helpfulness Prediction of Multimodal Reviews*
Junhao Liu, Zhen Hai, Min Yang and Lidong Bing

9:00–11:00 *Instantaneous Grammatical Error Correction with Shallow Aggressive Decoding*
Xin Sun, Tao Ge, Furu Wei and Houfeng Wang

9:00–11:00 *Automatic ICD Coding via Interactive Shared Representation Networks with Self-distillation Mechanism*
Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong and Shengping Liu

9:00–11:00 *PHMOSpell: Phonological and Morphological Knowledge Guided Chinese Spelling Check*
Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang and Jing Xiao

Poster 3O: Language Generation

9:00–11:00 *Guiding the Growth: Difficulty-Controllable Question Generation through Step-by-Step Rewriting*
Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin and Yefeng Zheng

9:00–11:00 *Improving Encoder by Auxiliary Supervision Tasks for Table-to-Text Generation*
Liang Li, Can Ma, Yinliang Yue and Dayong Hu

9:00–11:00 *POS-Constrained Parallel Decoding for Non-autoregressive Generation*
Kexin Yang, Wenqiang Lei, Dayiheng Liu, Weizhen Qi and Jiancheng Lv

9:00–11:00 *Bridging Subword Gaps in Pretrain-Finetune Paradigm for Natural Language Generation*
Xin Liu, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Min Zhang, Haiying Zhang and Jinsong Su

9:00–11:00 *TGEA: An Error-Annotated Dataset and Benchmark Tasks for TextGeneration from Pretrained Language Models*
Jie He, Bo Peng, Yi Liao, Qun Liu and Deyi Xiong

9:00–11:00 *Addressing Semantic Drift in Generative Question Answering with Auxiliary Extraction*
Chenliang Li, Bin Bi, Ming Yan, Wei Wang and Songfang Huang

Poster 3P: Summarization

9:00–11:00 *Long-Span Summarization via Local Attention and Content Selection*
Potsawee Manakul and Mark Gales

9:00–11:00 *RepSum: Unsupervised Dialogue Summarization based on Replacement Strategy*
Xiyan Fu, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Changlong Sun and Zhenglu Yang

9:00–11:00 *BASS: Boosting Abstractive Summarization with Unified Semantic Graph*
Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu and Haifeng Wang

9:00–11:00 *Capturing Relations between Scientific Papers: An Abstractive Model for Related Work Section Generation*
Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao and Rui Yan

9:00–11:00 *Focus Attention: Promoting Faithfulness and Diversity in Summarization*
Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe and Ryan McDonald

9:00–11:00 *Generating Query Focused Summaries from Query-Free Resources*
Yumo Xu and Mirella Lapata

9:00–11:00 *Demoting the Lead Bias in News Summarization via Alternating Adversarial Learning*
Linzi Xing, Wen Xiao and Giuseppe Carenini

Poster 3Q: Question Answering

9:00–11:00 *DuReader_robust: A Chinese Dataset Towards Evaluating Robustness and Generalization of Machine Reading Comprehension in Real-World Applications*
Hongxuan Tang, Hongyu Li, Jing Liu, Yu Hong, Hua Wu and Haifeng Wang

9:00–11:00 *Sequence to General Tree: Knowledge-Guided Geometry Word Problem Solving*
Shih-hung Tsai, Chao-Chun Liang, Hsin-Min Wang and Keh-Yih Su

9:00–11:00 *Robustifying Multi-hop QA through Pseudo-Evidentiality Training*
Kyungjae Lee, Seung-won Hwang, Sang-eun Han and Dohyeon Lee

9:00–11:00 *Multi-Scale Progressive Attention Network for Video Question Answering*
Zhicheng Guo, Jiaxuan Zhao, Licheng Jiao, Xu Liu and Lingling Li

9:00–11:00 *Efficient Passage Retrieval with Hashing for Open-domain Question Answering*
Ikuya Yamada, Akari Asai and Hannaneh Hajishirzi

9:00–11:00 *xMoCo: Cross Momentum Contrastive Learning for Open-Domain Question Answering*
Nan Yang, Furu Wei, Binxing Jiao, Daxing Jiang and Linjun Yang

9:00–11:00 *Learn to Resolve Conversational Dependency: A Consistency Training Framework for Conversational Question Answering*
Gangwoo Kim, Hyunjae Kim, Jungsoo Park and Jaewoo Kang

Poster 3R: Language Grounding to Vision, Robotics and Beyond

9:00–11:00 *PhotoChat: A Human-Human Dialogue Dataset With Photo Sharing Behavior For Joint Image-Text Modeling*
Xiaoxue Zang, Lijuan Liu, Maria Wang, Yang Song, Hao Zhang and Jindong Chen

9:00–11:00 *Good for Misconceived Reasons: An Empirical Revisiting on the Need for Visual Context in Multimodal Machine Translation*
Zhiyong Wu, Lingpeng Kong, Wei Bi, Xiang Li and Ben Kao

9:00–11:00 *Attend What You Need: Motion-Appearance Synergistic Networks for Video Question Answering*
Ahjeong Seo, Gi-Cheon Kang, Joonhan Park and Byoung-Tak Zhang

9:00–11:00 *Decoupling the Role of Data, Attention, and Losses in Multimodal Transformers*
Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac and Aida Nematzadeh

Poster 3S: Information Extraction

9:00–11:00 *BERTifying the Hidden Markov Model for Multi-Source Weakly Supervised Named Entity Recognition*
Yinghao Li, Pranav Shetty, Lucas Liu, Chao Zhang and Le Song

9:00–11:00 *CIL: Contrastive Instance Learning Framework for Distantly Supervised Relation Extraction*
Tao Chen, Haizhou Shi, Siliang Tang, Zhigang Chen, Fei Wu and Yueting Zhuang

9:00–11:00 *SENT: Sentence-level Distant Relation Extraction via Negative Training*
Ruotian Ma, Tao Gui, Linyang Li, Qi Zhang, Xuanjing Huang and Yaqian Zhou

9:00–11:00 *An End-to-End Progressive Multi-Task Learning Framework for Medical Named Entity Recognition and Normalization*
Baohang Zhou, Xiangrui Cai, Ying Zhang and Xiaojie Yuan

9:00–11:00 *PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction*
Hengyi Zheng, rui wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming and Yefeng Zheng

9:00–11:00 *Learning from Miscellaneous Other-Class Words for Few-shot Named Entity Recognition*
Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou and Juanzi Li

Wednesday, August 4, 2021 (all times UTC+0) (continued)

9:00–11:00 *Joint Biomedical Entity and Relation Extraction with Knowledge-Enhanced Collective Inference*
Tuan Lai, Heng Ji, ChengXiang Zhai and Quan Hung Tran

9:00–11:00 *Entity Concept-enhanced Few-shot Relation Extraction*
Shan Yang, Yongfei Zhang, Guanglin Niu, Qinghua Zhao and Shiliang Pu

9:00–11:00 *Fine-grained Information Extraction from Biomedical Literature based on Knowledge-enriched Abstract Meaning Representation*
Zixuan Zhang, Nikolaus Parulian, Heng Ji, Ahmed Elsayed, Skatje Myers and Martha Palmer

9:00–11:00 *Unleash GPT-2 Power for Event Detection*
Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt and Thien Huu Nguyen

9:00–11:00 *Improving Model Generalization: A Chinese Named Entity Recognition Case Study*
Guanqing Liang and Cane Wing-Ki Leung

9:00–11:00 *CLEVE: Contrastive Pre-training for Event Extraction*
Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li and Jie Zhou

9:00–11:00 *Three Sentences Are All You Need: Local Path Enhanced Document Relation Extraction*
Quzhe Huang, Shengqi Zhu, Yansong Feng, Yuan Ye, Yuxuan Lai and Dongyan Zhao

9:00–11:00 *Document-level Event Extraction via Parallel Prediction Networks*
Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao and Taifeng Wang

9:00–11:00 *StructuralLM: Structural Pre-training for Form Understanding*
Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang and Luo Si

Poster 3T: Sentiment Analysis, Stylistic Analysis, and Argument Mining

9:00–11:00 *Dual Graph Convolutional Networks for Aspect-based Sentiment Analysis*
Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie WANG and Eduard Hovy

9:00–11:00 *Multi-Label Few-Shot Learning for Aspect Category Detection*
Mengting Hu, Shiwan Zhao, Honglei Guo, Chao Xue, Hang Gao, Tiegang Gao, renhong cheng and Zhong Su

9:00–11:00 *Argument Pair Extraction via Attention-guided Multi-Layer Multi-Cross Encoding*
Liying Cheng, Tianyu Wu, Lidong Bing and Luo Si

9:00–11:00 *A Neural Transition-based Model for Argumentation Mining*
Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du and Ruifeng Xu

11:00–12:00 ***Lifetime Award***

Session 14A: Language Generation 2

14:00–14:10 *Keep It Simple: Unsupervised Simplification of Multi-Paragraph Text*
Philippe Laban, Tobias Schnabel, Paul Bennett and Marti A. Hearst

14:10–14:20 *Long Text Generation by Modeling Sentence-Level and Discourse-Level Coherence*
Jian Guan, Xiaoxi Mao, changjie fan, Zitao Liu, Wenbiao Ding and Minlie Huang

14:20–14:30 *OpenMEVA: A Benchmark for Evaluating Open-ended Story Generation Metrics*
Jian Guan, Zhexin Zhang, Zhuoer Feng, Zitao Liu, Wenbiao Ding, Xiaoxi Mao, changjie fan and Minlie Huang

14:30–14:40 *DYPLOC: Dynamic Planning of Content Using Mixed Language Models for Text Generation*
Xinyu Hua, Ashwin Sreevatsa and Lu Wang

14:40–14:50 *Controllable Open-ended Question Generation with A New Question Type Ontology*
Shuyang Cao and Lu Wang

14:50–15:00 *BERTGen: Multi-task Generation through BERT*
Faidon Mitzalis, Ozan Caglayan, Pranava Madhyastha and Lucia Specia

Session 14B: Machine Translation and Multilinguality 9

14:00–14:10 *Selective Knowledge Distillation for Neural Machine Translation*
Fusheng Wang, Jianhao Yan, Fandong Meng and Jie Zhou

14:10–14:20 *Measuring and Increasing Context Usage in Context-Aware Machine Translation*
Patrick Fernandes, Kayo Yin, Graham Neubig and André F. T. Martins

14:20–14:30 *Beyond Offline Mapping: Learning Cross-lingual Word Embeddings through Context Anchoring*
Aitor Ormazabal, Mikel Artetxe, Aitor Soroa, Gorka Labaka and Eneko Agirre

14:30–14:40 *CCMatrix: Mining Billions of High-Quality Parallel Sentences on the Web*
Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin and Angela Fan

14:40–14:50 *EDITOR: an Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints*
Weijia Xu and Marine Carpuat

14:50–15:00 *Gender Bias in Machine Translation*
Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri and Marco Turchi

Session 14C: Machine Learning for NLP 7

14:00–14:10 *Length-Adaptive Transformer: Train Once with Length Drop, Use Anytime with Search*
Gyuwan Kim and Kyunghyun Cho

14:10–14:20 *GhostBERT: Generate More Features with Cheap Operations for BERT*
Zhiqi Huang, Lu Hou, Lifeng Shang, Xin Jiang, Xiao Chen and Qun Liu

14:20–14:30 *Super Tickets in Pre-Trained Language Models: From Model Compression to Improving Generalization*
Chen Liang, Simiao Zuo, Minshuo Chen, Haoming Jiang, Xiaodong Liu, Pengcheng He, Tuo Zhao and Weizhu Chen

14:30–14:40 *A Novel Estimator of Mutual Information for Learning to Disentangle Textual Representations*
Pierre Colombo, Pablo Piantanida and Chloé Clavel

14:40–14:50 *Determinantal Beam Search*
Clara Meister, Martina Forster and Ryan Cotterell

14:50–15:00 *Multi-hop Graph Convolutional Network with High-order Chebyshev Approximation for Text Reasoning*
Shuoran Jiang, Qingcai Chen, Xin Liu, Baotian Hu and Lisai Zhang

Session 14D: NLP Applications 4

14:00–14:10 *Accelerating Text Communication via Abbreviated Sentence Input*
Jiban Adhikary, Jamie Berger and Keith Vertanen

14:10–14:20 *Regression Bugs Are In Your Model! Measuring, Reducing and Analyzing Regressions In NLP Model Updates*
YUQING XIE, Yi-An Lai, Yuanjun Xiong, Yi Zhang and Stefano Soatto

14:20–14:30 *Detecting Propaganda Techniques in Memes*
Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov and Giovanni Da San Martino

14:30–14:37 *Unsupervised Cross-Domain Prerequisite Chain Learning using Variational Graph Autoencoders*
Irene Li, Vanessa Yan, Tianxiao Li, Rihao Qu and Dragomir Radev

14:37–14:44 *Attentive Multiview Text Representation for Differential Diagnosis*
Hadi Amiri, Mitra Mohtarami and Isaac Kohane

14:44–14:51 *MedNLI Is Not Immune: Natural Language Inference Artifacts in the Clinical Domain*
Christine Herlihy and Rachel Rudinger

Session 14E: Question Answering 4

14:00–14:10 *On the Efficacy of Adversarial Data Collection for Question Answering: Results from a Large-Scale Randomized Study*
Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton and Wen-tau Yih

14:10–14:20 *Learning Dense Representations of Phrases at Scale*
Jinhyuk Lee, Mujeen Sung, Jaewoo Kang and Danqi Chen

14:20–14:30 *End-to-End Training of Neural Retrievers for Open-Domain Question Answering*
Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton and Bryan Catanzaro

14:30–14:40 *Question Answering Over Temporal Knowledge Graphs*
Apoorv Saxena, Soumen Chakrabarti and Partha Talukdar

14:40–14:47 *Towards a more Robust Evaluation for Conversational Question Answering*
Wissam Siblini, Baris Sayil and Yacine Kessaci

14:47–14:54 *VAULT: VAriable Unified Long Text Representation for Machine Reading Comprehension*
Haoyang Wen, Anthony Ferritto, Heng Ji, Radu Florian and Avi Sil

Session 15A: Language Generation 3

15:00–15:10 *Language Model Augmented Relevance Score*
Ruibo Liu, Jason Wei and Soroush Vosoughi

15:10–15:20 *DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts*
Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith and Yejin Choi

15:20–15:30 *Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models*
Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer and Daniel Weld

15:30–15:40 *Metaphor Generation with Conceptual Mappings*
Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan and Iryna Gurevych

15:40–15:50 *Computational Framework for Slang Generation*
Zhewei Sun, Richard Zemel and Yang Xu

15:50–15:57 *Avoiding Overlap in Data Augmentation for AMR-to-Text Generation*
Wenchao Du and Jeffrey Flanigan

Session 15B: NLP Applications 5

15:00–15:10 *Learning Latent Structures for Cross Action Phrase Relations in Wet Lab Protocols*
Chaitanya Kulkarni, Jany Chan, Eric Fosler-Lussier and Raghu Machiraju

15:10–15:20 *Multimodal Multi-Speaker Merger & Acquisition Financial Modeling: A New Task, Dataset, and Neural Baselines*
Ramit Sawhney, Mihir Goyal, Prakhar Goel, Puneet Mathur and Rajiv Ratn Shah

15:20–15:30 *Mid-Air Hand Gestures for Post-Editing of Machine Translation*
Rashad Albo Jamara, Nico Herbig, Antonio Krüger and Josef van Genabith

15:30–15:40 *Inter-GPS: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning*
Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang and Song-Chun Zhu

15:40–15:50 *Joint Verification and Reranking for Open Fact Checking Over Tables*
Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih and Sebastian Riedel

15:50–15:57 *Weakly-Supervised Methods for Suicide Risk Assessment: Role of Related Domains*
Chenghao Yang, Yudong Zhang and Smaranda Muresan

Session 15C: Resources and Evaluation 5

15:00–15:10 *Evaluation of Thematic Coherence in Microblogs*
Iman Munire Bilal, Bo Wang, Maria Liakata, Rob Procter and Adam Tsakalidis

15:10–15:20 *Neural semi-Markov CRF for Monolingual Word Alignment*
Wuwei Lan, Chao Jiang and Wei Xu

15:20–15:30 *Privacy at Scale: Introducing the PrivaSeer Corpus of Web Privacy Policies*
Mukund Srinath, Shomir Wilson and C Lee Giles

15:30–15:40 *The statistical advantage of automatic NLG metrics at the system level*
Johnny Wei and Robin Jia

15:40–15:50 *Are Missing Links Predictable? An Inferential Benchmark for Knowledge Graph Completion*
Yixin Cao, Xiang Ji, Xin Lv, Juanzi Li, Yonggang Wen and Hanwang Zhang

15:50–15:57 *Can Transformer Models Measure Coherence In Text: Re-Thinking the Shuffle Test*
Philippe Laban, Luke Dai, Lucas Bandarkar and Marti A. Hearst

Session 15D: Summarization 2

15:00–15:10 *ConvoSumm: Conversation Summarization Benchmark and Improved Abstractive Summarization with Argument Mining*
Alexander Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad and Dragomir Radev

15:10–15:20 *Improving Factual Consistency of Abstractive Summarization via Question Answering*
Feng Nan, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Kathleen McKeown, Ramesh Nallapati, Dejiao Zhang, Zhiguo Wang, Andrew O. Arnold and Bing Xiang

15:20–15:30 *EmailSum: Abstractive Email Thread Summarization*
Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao and Mohit Bansal

15:30–15:40 *Cross-Lingual Abstractive Summarization with Limited Parallel Resources*
Yu Bai, Yang Gao and Heyan Huang

15:40–15:50 *Dissecting Generation Modes for Abstractive Summarization Models via Ablation and Attribution*
Jiacheng Xu and Greg Durrett

15:50–15:57 *SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization*
Yixin Liu and Pengfei Liu

Session 15E: Semantics: Lexical Semantics 2

15:00–15:10 *Learning Prototypical Functions for Physical Artifacts*
Tianyu Jiang and Ellen Riloff

15:10–15:20 *Verb Knowledge Injection for Multilingual Event Processing*
Olga Majewska, Ivan Vulić, Goran Glavaš, Edoardo Maria Ponti and Anna Korhonen

15:20–15:30 *Dynamic Contextualized Word Embeddings*
Valentin Hofmann, Janet Pierrehumbert and Hinrich Schütze

15:30–15:40 *Lexical Semantic Change Discovery*
Sinan Kurtyigit, Maike Park, Dominik Schlechtweg, Jonas Kuhn and Sabine Schulte im Walde

15:40–15:50 *Analysis and Evaluation of Language Models for Word Sense Disambiguation*
Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar and Jose Camacho-Collados

15:50–16:00 *Let's Play mono-poly: BERT Can Reveal Words' Degree of Polysemy*
Aina Garí Soler and Marianna Apidianaki

Session 16A: Dialog and Interactive Systems 7

16:00–16:10 *Pretraining the Noisy Channel Model for Task-Oriented Dialogue*
Qi Liu, Lei Yu, Laura Rimell and Phil Blunsom

16:10–16:20 *The R-U-A-Robot Dataset: Helping Avoid Chatbot Deception by Detecting User Questions About Human or Non-Human Identity*
David Gros, Yu Li and Zhou Yu

16:20–16:30 *Conversation Graph: Data Augmentation, Training and Evaluation for Non-Deterministic Dialogue Management*
Milan Gritta, Gerasimos Lampourasm and Ignacio Iacobacci

16:30–16:40 *Using Meta-Knowledge Mined from Identifiers to Improve Intent Recognition in Conversational Systems*
Claudio Pinhanez, Paulo Cavalin, Victor Henrique Alves Ribeiro, Ana Appel, Heloisa Candello, Julio Nogima, Mauro Pichiliani, Melina Guerra, Maira de Bayser, Gabriel Malfatti and Henrique Ferreira

16:40–16:50 *Space Efficient Context Encoding for Non-Task-Oriented Dialogue Generation with Graph Attention Transformer*
Fabian Galetzka, Jewgeni Rose, David Schlangen and Jens Lehmann

16:50–17:00 *DialogueCRN: Contextual Reasoning Networks for Emotion Recognition in Conversations*
Dou Hu, Lingwei Wei and Xiaoyong Huai

Session 16B: Resources and Evaluation 6

16:00–16:10 *Cross-replication Reliability - An Empirical Approach to Interpreting Inter-rater Reliability*
Ka Wong, Praveen Paritosh and Lora Aroyo

16:10–16:20 *TIMEDIAL: Temporal Commonsense Reasoning in Dialog*
Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi and Manaal Faruqui

16:20–16:30 *RAW-C: Relatedness of Ambiguous Words in Context (A New Lexical Resource for English)*
Sean Trott and Benjamin Bergen

16:30–16:40 *ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic*
Muhammad Abdul-Mageed, AbdelRahim Elmadany and El Moatez Billah Nagoudi

16:40–16:47 *SaRoCo: Detecting Satire in a Novel Romanian Corpus of News Articles*
Ana-Cristina Rogoz, Gaman Mihaela and Radu Tudor Ionescu

16:47–16:54 *Bringing Structure into Summaries: a Faceted Summarization Dataset for Long Scientific Documents*
Rui Meng, khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang and Daqing He

Session 16C: Semantics: Sentence-level Semantics, Textual Inference and Other areas 4

16:00–16:10 *Improving Paraphrase Detection with the Adversarial Paraphrasing Task*
Animesh Nighojkar and John Licato

16:10–16:20 *ADEPT: An Adjective-Dependent Plausibility Task*
Ali Emami, Ian Porada, Alexandra Olteanu, Kaheer Suleman, Adam Trischler and Jackie Chi Kit Cheung

16:20–16:30 *ReadOnce Transformers: Reusable Representations of Text for Transformers*
Shih-Ting Lin, Ashish Sabharwal and Tushar Khot

16:30–16:40 *Conditional Generation of Temporally-ordered Event Sequences*
Shih-Ting Lin, Nathanael Chambers and Greg Durrett

16:40–16:50 *Hate Speech Detection Based on Sentiment Knowledge Sharing*
Xianbing Zhou, yang yong, xiaochao fan, Ge Ren, Yunfeng Song, Yufeng Diao, Liang Yang and Hongfei LIN

Session 16D: Syntax: Tagging, Chunking, and Parsing 2

16:00–16:10 *Transition-based Bubble Parsing: Improvements on Coordination Structure Prediction*
Tianze Shi and Lillian Lee

16:10–16:20 *SpanNER: Named Entity Re-/Recognition as Span Prediction*
Jinlan Fu, Xuanjing Huang and Pengfei Liu

16:20–16:30 *Strong Equivalence of TAG and CCG*
Lena Katharina Schiffer and Andreas Maletti

16:30–16:40 *StructFormer: Joint Unsupervised Induction of Dependency and Constituency Structure from Masked Language Modeling*
Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler and Aaron Courville

16:40–16:47 *Replicating and Extending "Because Their Treebanks Leak": Graph Isomorphism, Covariants, and Parser Performance*
Mark Anderson, Anders Søgaard and Carlos Gómez-Rodríguez

Session 16E: Machine Translation and Multilinguality 10

16:00–16:10 *Language Embeddings for Typology and Cross-lingual Transfer Learning*
Dian Yu, Taiqi He and Kenji Sagae

16:10–16:20 *Can Sequence-to-Sequence Models Crack Substitution Ciphers?*
Nada Aldarrab and Jonathan May

16:20–16:30 *Beyond Noise: Mitigating the Impact of Fine-grained Semantic Divergences on Neural Machine Translation*
Eleftheria Briakou and Marine Carpuat

16:30–16:40 *Revisiting Negation in Neural Machine Translation*
Gongbo Tang, Philipp Rönchen, Rico Sennrich and Joakim Nivre

16:40–16:50 *Discriminative Reranking for Neural Machine Translation*
Ann Lee, Michael Auli and Marc'Aurelio Ranzato

16:50–16:57 *Don't Rule Out Monolingual Speakers: A Method For Crowdsourcing Machine Translation Data*
Rajat Bhatnagar, Ananya Ganesh and Katharina Kann

Best Paper Session

23:00–23:03 *EXPLAINABOARD: An Explainable Leaderboard for NLP*
Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye and Graham Neubig

23:03–23:16 *Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering*
Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei and Christopher Manning

23:16–23:29 *All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text*
Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan and Noah A. Smith

23:29–23:42 *Scientific Credibility of Machine Translation Research: A Meta-Evaluation of 769 Papers*
Benjamin Marie, Atsushi Fujita and Raphael Rubino

23:42–23:55 *Neural Machine Translation with Monolingual Translation Memory*
Deng Cai, Yan Wang, Huayang Li, Wai Lam and Lemao Liu

23:55–00:08 *Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning*
Armen Aghajanyan, Sonal Gupta and Luke Zettlemoyer

00:08–00:21 *UnNatural Language Inference*
Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau and Adina Williams

00:21–00:39 *Including Signed Languages in Natural Language Processing*
Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg and Malihe Alikhani

00:39–00:57 *Vocabulary Learning via Optimal Transport for Neural Machine Translation*
Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng and Lei Li

Thursday, August 5, 2021 (all times UTC+0)

01:00–01:30 ***Distinguished Service and Test-Of-Time Awards session***

01:30–02:00 ***Closing and Future Conferences***

Catchphrase: Automatic Detection of Cultural References

Nir Sweed
The Hebrew University of Jerusalem
nir.sweed@mail.huji.ac.il

Dafna Shahaf
The Hebrew University of Jerusalem
dshahaf@cs.huji.ac.il

Abstract

A *snowclone* is a customizable phrasal template that can be realized in multiple, instantly recognized variants. For example, "* is the new *" (Orange is the new black, 40 is the new 30). Snowclones are extensively used in social media. In this paper, we study snowclones originating from pop-culture quotes; our goal is to automatically detect cultural references in text. We introduce a new, publicly available data set of pop-culture quotes and their corresponding snowclone usages and train models on them. We publish code for CATCHPHRASE, an internet browser plugin to automatically detect and mark references in real-time, and examine its performance via a user study. Aside from assisting people to better comprehend cultural references, we hope that detecting snowclones can complement work on paraphrasing and help to tackle long-standing questions in social science about the dynamics of information propagation.

1 Introduction

First coined by Richard Dawkins (Dawkins, 1976), a meme is a unit of cultural transmission: any idea or behavior that can be transferred by imitation. Internet memes have become an integral part of modern digital culture (Shifman, 2014). Pullum (Pullum, 2004) coined the term *snowclones* to describe a specific type of meme – phrasal templates that are easily reusable in many different contexts. Pullum described a snowclone as "a multi-use, customizable, instantly recognizable, time-worn, quoted or misquoted phrase or sentence that can be used in an entirely open array of different jokey variants". For example, the quote "One does not simply walk into Mordor" from the "Lord of the Rings" films became a well-known pattern – *"One does not simply *"* – used extensively online (see Figure 1).

In this paper, our goal is to develop algorithms to **detect snowclones** in text. We envision an "Englishman in New York" – a foreigner, perhaps, or someone who does not easily understand contemporary cultural references and could use the help of an automated system to communicate better. In particular, we focus on pop-culture references over the internet.

From a linguistic point of view, snowclones complement the paraphrasing task (Barzilay and McKeown, 2001; Fernando and Stevenson, 2008; Dolan et al., 2004). Paraphrase detection identifies alternative ways to convey the same meaning, while snowclones keep (some of) the original sentence structure but completely change the meaning.

Detection and tracking of digital memes have been the focus of multiple computational studies. The closest to our work are MEMETRACKER and NIFTY (Leskovec et al., 2009; Suen et al., 2013), that tracked quotations attributed to individuals. These works focused on short, distinctive phrases that travel *relatively intact* through on-line text. Other related tasks are multi-word expression/idiom identification (Haagsma et al., 2020; Zarrieß and Kuhn, 2009; Muzny and Zettlemoyer, 2013) and cliché detection (Cook and Hirst, 2013; van Cranenburgh, 2018). Again, idioms and multi-word expressions are chiefly fixed expressions ("cat got your tongue?", "jumped the shark") that rarely change their meaning across mutations. Therefore,

Figure 1: Snowclone example, based on "One does not simply walk into Mordor" from "Lord of the Rings".

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1–7
August 1–6, 2021. ©2021 Association for Computational Linguistics

these settings are much more restrictive than ours.

Our contributions are the following: we propose a novel task of snowclone detection, identifying cultural references. We first formulate it as a tagging task, treating snowclones as regular expressions; we conduct a user study to show humans have an intuitive notion of the "correct" pattern(s), and develop a sequence-to-sequence tagger to reveal such patterns. We then extend the formulation to softer notions of similarity. We experiment with feature-based and neural approaches, achieving high accuracies. To further show the utility of our methods, we develop CATCHPHRASE, a browser extension to detect pop-culture references, conduct a user study and show it indeed helps users identify cultural references. We publish data and code[1]. We believe tracking snowclones will find interesting applications in social science, exploring the diffusion and evolution of highly dynamic content online.

2 Snowclones as Regular Expressions

The common view of snowclones treats them as regular expressions (The-Snowclones-Database, 2007). Thus, in this section, we formulate the snowclone detection problem as a tagging task. Intuitively, we want to predict for each word in the original sentence whether it is replaced by a wildcard. We use the resulting pattern to match new sentences to the original sentence. For example, given a sentence $s = \langle One, does, not, simply, walk, into, Mordor \rangle$ we would like to find a mapping: $T(s) = \langle One, does, not, simply, *, *, * \rangle$. (Adjacent wildcards can be merged)

2.1 Can People do This?

Before we set out to find an algorithm to uncover the underlying snowclone form of an input sentence, we try to evaluate the feasibility of this task. It is not clear that such patterns exist, or are agreed upon by human annotators. To that end, we conduct a user study to test if people have an intuitive notion of snowclone patterns.

We recruited 22 volunteers through social media. The participants were 80% males. 85% of them were 25-35 years old, the rest being 40-55. All participants were Israeli and identified as non-native English speakers. Participants were given a short explanation of snowclones and instructed to find the snowclone form of a set of (the same) 20 sentences, chosen from the memorable movie quote database (Danescu-Niculescu-Mizil et al., 2012). We chose sentences at random, filtering out quotes that became known internet memes. Participants marked words that should become wildcards, generating up to 3 patterns per sentence, as they saw fit. We asked participants to report whether they were familiar with any sentence, and discarded the entire questionnaire of two who did.

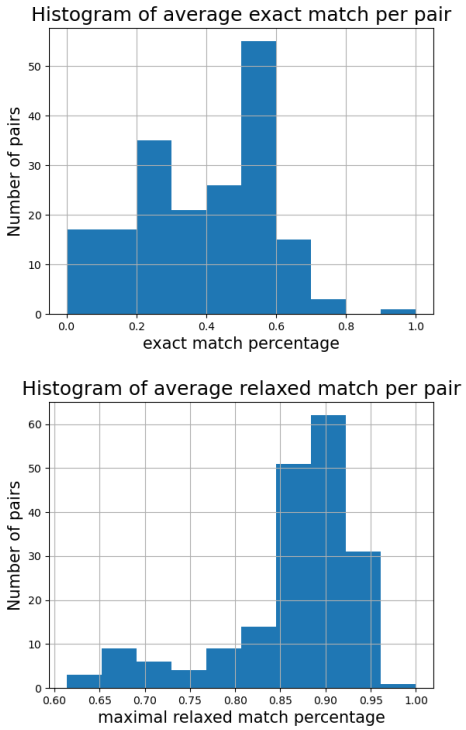


Figure 2: Histogram of the exact-match similarity measure (top) and relaxed-match measure (bottom), averaged over all sentences, for all pairs of participants.

Evaluation and Results. To compute similarity between pairs of people, we propose two measures. In **exact-match**, the score is the percentage of sentences (out of 20) on which the two people had at least one exact-match pattern. In **relaxed-match**, we compute for each sentence the closest match between the patterns of both people (in terms of simple % agreement). The score is the percentage of agreement over all 20 closest matches.

Figure 2 shows histograms over all pairs of participants. For exact-match, most pairs of participants agree on roughly half the patterns. A careful examination of the results indicates that participants are divided into those that prefer a single, general pattern (annotating "The pavement was his enemy" as "The * was his *"), and those preferring

[1] https://github.com/sweedy12/CATCHPHRASE

Snowclone Form Tagging - Results		
Model	Accuracy	Recall
Naive	0.74	0
Bi-LSTM-CRF	**0.92**	0.82
BERT	0.9	**0.88**

Table 1: Accuracy and recall for each of the proposed models for the snowclone form tagging task.

several narrower patterns (marking both "The * was his enemy" and "The pavement was his *"). Another contributing factor is that many mismatched pairs of patterns differ only in stopwords. The relaxed-match measure is less sensitive to this issue and indeed demonstrates high agreement. We hypothesize that this indicates the feasibility of training machine learning models for this task.

3 Snowclone Pattern Tagger

We create and publish our own data set for this task, and use it to train two different ML models for it.

Data. To train ML models to solve the task of snowclone tagging, we needed examples for sentences and their underlying snowclone form. To this end, we use the snowclone patterns along with the original quotes from The Snowclone Database (The-Snowclones-Database, 2007). As this is not enough data to train on, we use the patterns to lookup more instances online, collecting 7700 $\langle$*snowclone pattern, instance*$\rangle$ pairs. When splitting to train-dev-test sets (60%/20%/20%), we make sure all variants of the same pattern are put in the same set. We release our dataset[1].

Bi-LSTM-CRF. We adapt the model of (Huang et al., 2015), tested on part of speech tagging, chunking and named entity recognition tasks. Its CRF layer performs a structured prediction over the sentence tags, using sentence-level information rather than predicting a label for each word separately, rendering it useful for our task. For optimization, we use negative log-likelihood.

BERT S2S. We use BERT (Devlin et al., 2019), as it has shown to produce good results when fine-tuned to specific sequence-to-sequence tasks. We fine-tune BERT for a token classification task using the snowclone form dataset. Since this model outputs a probability measure for each token, we use binary cross-entropy as the objective function.

See Appendix A for implementation details and hyper-parameter tuning.

Evaluation and results. Since most words in an input sentence are not replaceable, wildcards are infrequent. Thus, we prioritize models with higher recall than precision. Table 1 shows recall and accuracy of the models. The naive majority baseline (no words are wildcards) yields 74% accuracy (and, naturally, 0% recall). The Bi-LSTM-CRF model reaches an accuracy of 92%, and 82% recall. BERT achieves an accuracy of 90%, and recall 88%.

4 Going Beyond Regular Expressions

When we tried to apply our models to find snowclones in online community text (looking for regex matches), we realized that the regex formulation might be too simplistic, as some cultural references do not follow the snowclone pattern exactly, and some sentences that do follow it are not really references. Take Apocalypse Now's famous "I love the smell of napalm in the morning". A natural corresponding pattern is "I love the smell of * in the morning", and indeed, "I love the smell of bureaucracy in the morning" is most likely a reference to the movie. However, the case of "I love the smell of *pancakes* in the morning" is a lot less clear. On the other hand, "30 is the old 40" does not perfectly match the "* is the new *" pattern, but still might be considered a reference. In this section we reformulate the problem, using the output of the sequence-to-sequence tagger as one input to a machine learning model.

We reformulate the problem as a binary classification task over pairs of sentences. Given a seed sentence s representing an original pop-culture quote, and a candidate sentence c, decide whether c is a reference to s. We note this is not an easy task, as it is hard to put our finger on why "One does not simply forget to social distance" is likely a reference to "One does not simply walk into Mordor", but "One cannot just walk right into jail" is not.

5 Snowclone Reference Detector

Data. We searched the web and found 20 famous movie quotes that turned into snowclone internet memes. We removed three quotes appearing in the data of Section 3, not to contaminate our evaluation. Next, we defined overly general regular expressions for each seed (attempting to catch both snowclones and not) and crawled Reddit conversations to find matches. We choose Reddit due to its popularity and comprehensive use of memes. We collected 3850 pairs of seed and sentence and had

Snowclone Detection - Results			
Model	Accuracy	Precision	Recall
Naive	0.64	1	0
SVM	**0.85±0.08**	**0.84±0.13**	**0.78±0.12**
RoBERTa	0.81±0.94	0.7± 0.15	0.74±0.18

Table 2: Snowclone detection task. We performed 20 splits for the SVM model and 5 for RoBERTa, and report standard deviation.

an expert manually annotate them (after calibration). The dataset is imbalanced, with 64% of pairs tagged as non-reference. When splitting to train-dev-test (60%/20%/20%), we ensure all examples from the same seed are put in the same set. We take a supervised approach and train two models.

Feature-based SVM model. We calculate three sets of features, focusing on sentence *structure*. (1) Similarity between s and c: edit distance, longest common sequence, and longest substring between s, c. (2) We use the snowclone tagger of Section 3 to predict $\hat{s}$, the snowclone form of s and use the same features of group (1) between $\hat{s}, c$. (3) To characterize the shared and replaced words we calculate the idf statistic for words shared between s, c and words in s but not in c (idf over movie quotes (Danescu-Niculescu-Mizil and Lee, 2011)). We tried decision trees, random forests, and SVM, and chose SVM due to its performance.

RoBERTa-based model. We chose RoBERTa as our second model, as it showed impressive results on a related 2-sentence classification task. We use a model pre-trained on SNLI (Nie et al., 2020), which achieved state-of-the-art result on a natural language inference task. We replace its classification head with a binary classification head, and fine-tune the model on the dataset of Section 4. Unlike SVM, we expect this model to capture semantic similarity (e.g., between "old" and "new").

See Appendix B for implementation details and hyper-parameter tuning.

Evaluation and results. The accuracy, precision and recall measures for all models are presented in Table 2. The naive majority baseline achieves 64% accuracy on the full data set (as the data is not balanced). For our feature-based SVM model, we randomly select 20 different splits, reaching an average of 85% accuracy, 84% precision and 78% recall, with a corresponding std of 8.7%, 13.8% and 12%. The RoBERTa-based model achieved average results of 81% accuracy, 70% precision and 74% recall, with std 9.4%, 15.7% and 18.7%. Thus, we chose the SVM model. This perhaps indicates the importance of structure in the snowclone problem; alternatively, perhaps the amount of data was not sufficient to fine-tune RoBERTa.

Observations. As a (qualitative) reality check, we choose 10 seeds unseen during training. We crawl all Reddit posts from March 2016 (month and year chosen at random). We choose Reddit as a diverse and popular online community, where internet memes are used regularly. We use the SVM model to collect new candidate references for the seeds. We analyze the candidate references and observe that (not surprisingly) their quality is heavily influenced by the snowclone tagger feature. When the regex is too general (e.g., "I am your *" for "I am your father"), the number of false positives is high. Importantly, over all seeds our method is capable of detecting true references that do not exactly match the predicted snowclone-form.

6 Evaluation: Web Extension

Our main motivation in this study was to help the proverbial "Englishman in New York" identify cultural references. In this section, we ask whether *our algorithms can help users detect pop-culture references online*. We create CATCHPHRASE, a web browser extension able to detect and mark pop-culture references in web pages (see Figure 3). The extension inspects the web page source and identifies candidate sentences. We use locality-sensitive hashing (Gionis et al., 1999) with similarity threshold = 0.2 for filtering, allowing us to reduce computation time and maintain a small number of false negatives. Next, the extension runs the reference-detector on each (seed, candidate sentence) pair and highlights the predicted references.

Experimental design. We choose a set of 20 pop-culture quotes (seeds) unseen by our reference-detector during training time, and whose snowclone form is the basis to many variations. All sentences chosen are ones that became popular internet memes. For each seed quote, we manually crawled Reddit and found threads containing references to it. After filtering out threads that were over 10 messages long, we were left with 106 threads.

We recruited 10 volunteers through social media, all Israeli, non-native English speakers, who self-identified as having low familiarity with pop-culture. 80% of the volunteers were 20-35 years old, and the remaining 20% were 40-60 years old.

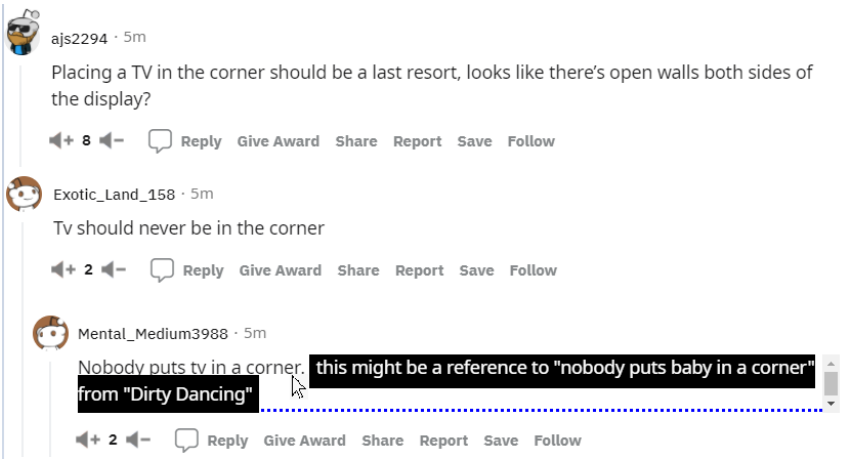


Figure 3: Screenshot of our web extension, suggesting "Nobody puts TV in a corner" is a reference to Dirty Dancing's "Nobody puts baby in a corner". The suggested reference is underlined in blue. Hovering over the underlined sentence prompts a message containing original quote information.

70% of the participants were females. We randomly selected 16 seeds for each participant and randomly split them into two groups, one per condition (with and without our extension). The threads were shown in random order. The participants were asked to go over each thread and point out any pop-culture references they detect, specifying their origin if they knew it.

Evaluation and results. Under the no-extension condition, participants correctly identified a pop-culture reference 38.7% of the time. The reference origin was correctly identified in 61.2% of these. This is interesting, as it shows people can identify that a sentence *looks* like a cultural reference, even when they do not recognize the source.

When using the extension, participants correctly identified a reference 68.7% of the times, recognizing the origin in 98.1% of these. In 26.3% of the threads, the algorithm did not recognize the reference. 5% of the times, we believe the algorithm was right but people thought it was not (e.g., "I solemnly swear I'm up for good tea" as a reference to "I solemnly swear I'm up to no good"). The reason source recognition is not perfect is one user finding a sentence the algorithm missed (but not attributing it). To check our hypothesis that web-extension users recognize more pop-culture references, we run t-test with $\alpha = 0.95$ and reject the null hypothesis with $pval = 0.00005$.

7 Conclusions and Future Work

In this work we proposed the novel task of detecting snowclones in text. Motivated by the high agreement achieved by humans on a snowclone annotation task, we first developed algorithms for finding snowclones which are regular expressions, then extended the formulation to a softer notion of similarity. We introduce a new data set of pop-culture quotes and their corresponding snowclone variants and train models on them. We publish code for CATCHPHRASE, an internet browser plugin to automatically detect and mark references in real-time. Our results demonstrate our algorithms can indeed help users detect pop-culture references.

In the future, our work might be used in conversational AI context, supporting agents' ability to understand and even generate pop-culture references. Another direction worth pursuing is applying our methods to domains outside pop-culture (or at the very least, to pop-culture of different cultures).

We believe snowclones, complementing the notion of paraphrases, are worth exploring and can give us new insights into how ideas spread and evolve. Our approach opens an opportunity to better answer long-standing questions in social science about the dynamics of information.

Acknowledgements

We thank the anonymous reviewers for their insightful and highly constructive comments and Roy Schwartz for stimulating discussions. This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 852686, SIAM), US National Science Foundation, US-Israel Binational Science Foundation (NSF-BSF) grant no. 2017741, and Amazon Research Awards.

References

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57, Toulouse, France. Association for Computational Linguistics.

Paul Cook and Graeme Hirst. 2013. Automatically assessing whether a text is cliched, with applications to literary analysis. In *Proceedings of the 9th Workshop on Multiword Expressions*, pages 52–57, Atlanta, Georgia, USA. Association for Computational Linguistics.

Andreas van Cranenburgh. 2018. Cliche expressions in literary and genre novels. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 34–43, Santa Fe, New Mexico. Association for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. 2012. You had me at hello: How phrasing affects memorability. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–901, Jeju Island, Korea. Association for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87, Portland, Oregon, USA. Association for Computational Linguistics.

Richard Dawkins. 1976. *The selfish gene*. Oxford university press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William Dolan, Chris Quirk, Chris Brockett, and Bill Dolan. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics*, pages 45–52.

Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529.

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 279–287, Marseille, France. European Language Resources Association.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506.

Grace Muzny and Luke Zettlemoyer. 2013. Automatic idiom identification in Wiktionary. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1421, Seattle, Washington, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Geoffrey Pullum. 2004. Snowclones: Lexicographical dating to the second.

Limor Shifman. 2014. *Memes in digital culture*. MIT press.

Caroline Suen, Sandy Huang, Chantat Eksombatchai, Rok Sosic, and Jure Leskovec. 2013. Nifty: a system for large scale information flow tracking and clustering. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1237–1248.

The-Snowclones-Database. 2007. The snowclone database.

Sina Zarrieß and Jonas Kuhn. 2009. Exploiting translational correspondences for pattern-independent MWE identification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009)*, pages 23–30, Singapore. Association for Computational Linguistics.

Below we provide implementation details for the sake of reproducibility.

A Snowclone Pattern Tagger: Hyper-parameter tuning

For the BI-LSTM-CRF model, we perform a small grid search to determine the values for the learning rate, weight decay, and the number of layers and hidden dimension of the the BI-LSTM. As the search space, we used $learning - rate \in \{0.01, 0.001, 0.0001\}$, *weight-decay* $\in \{0, 0.01, 0.001\}$, *num-layers* $\in \{1, 2, 3\}$ and *hidden-dim* $\in \{32, 64, 128\}$. Finally, we choose *learning-rate* $= 0.01$,*weight-decay* $= 0$, *num-layers* $= 2$,*hidden-dim* $= 32$. For the BERT model, we use a smaller grid search over the learning rate ($\in \{0.001, 0.0001\}$) and the weight decay ($\{0, 0.01, 0.001\}$) hyper-parameters, and train it for a single epoch using *learning-rate* $= 0.0001$,*weight-decay* $= 0.01$.

B Snowclone Reference Detector: Hyper-parameter tuning

For the RoBERTa model, we perform the same hyper-parameter search as described in Section A, and use the same values. For the SVM model, we search over kernels (RBF, linear and polynomial), degree (when applicable, over $[2, 3, 4]$) and C-values ($\{0.1 \cdot i\}_{i=1}^{10}$). Our search dictates using a polynomial kernel of degree 3, with $C = 0.5$.

On Training Instance Selection for Few-Shot Neural Text Generation

Ernie Chang[*], **Xiaoyu Shen**[*], **Hui-Syuan Yeh, Vera Demberg,**
Dept. of Language Science and Technology, Saarland University
{cychang,xshen}@coli.uni-saarland.de

Abstract

Large-scale pretrained language models have led to dramatic improvements in text generation. Impressive performance can be achieved by finetuning only on a small number of instances (few-shot setting). Nonetheless, almost all previous work simply applies random sampling to select the few-shot training instances. Little to no attention has been paid to the selection strategies and how they would affect model performance. In this work, we present a study on training instance selection in few-shot neural text generation. The selection decision is made based only on the unlabeled data so as to identify the most worthwhile data points that should be annotated under some budget of labeling cost. Based on the intuition that the few-shot training instances should be diverse and representative of the entire data distribution, we propose a simple selection strategy with K-means clustering. We show that even with the naive clustering-based approach, the generation models consistently outperform random sampling on three text generation tasks: data-to-text generation, document summarization and question generation. The code and training data are made available at https://gitlab.com/erniecyc/few-selector. We hope that this work will call for more attention on this largely unexplored area.

1 Introduction

Few-shot text generation is an important research topic since obtaining large-scale training data for each individual downstream task is prohibitively expensive. Recently, pretraining large neural networks with a language modeling objective has led to significant improvement across different few-shot text generation tasks (Radford et al., 2019; Lewis et al., 2020) and many techniques are proposed based on them (Chen et al., 2020; Schick and Schütze, 2020a; Zhang et al., 2020; Kale, 2020; Chang et al., 2020, 2021b; Li and Liang, 2021; Chang et al., 2021a). However, all the previous works simulate the few-shot scenario by randomly sampling a subset from the full training data. Little to no attention has been paid to the selection strategies.

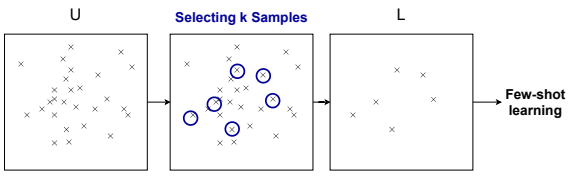


Figure 1: **Training scenario**: **U** represents unlabeled data and **L** indicates labeled instances. The annotation budget only allows selecting K data for annotating the reference text.

In this work, we present a preliminary study at searching for an optimal strategy to select the few-shot training instances. Studying the selection strategy is motivated by two rationales. First, random sampling leads to a large variance of model performance (Zhang et al., 2020; Schick and Schütze, 2020a,b). Yet current works sample their own training data which makes it difficult to compare across different models. One can then not be sure whether an improved performance can be really ascribed to the model or the randomness of sampling. Using a stable selection strategy to find the most informative few-shot instances can provide a fair platform and better benchmark different few-shot generative models. Second, in practical applications, e.g. document summarization, the training data is usually obtained by manually annotating the summaries for some selected documents. In Figure 1, we illustrate the typical training scenario for text generation where the annotation budget only allows annotating a limited amount of data. Studying the optimal selection strategy can help make the most use of our annotation budget. Specifically, we focus on

[*]Equal contribution. X.shen is now at Amazon Alexa AI.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 8–13
August 1–6, 2021. ©2021 Association for Computational Linguistics

the label-free setting where *the selection can only condition on the unannotated data*. Although leveraging the reference text may benefit the selection strategy, it conflicts with the realistic setting where we need to first select the data then get its annotated reference text.

The selection task resembles the theme of active learning (Balcan et al., 2007), where the model keeps identifying the most informative instances to get labeled. Existing active learning approaches can be roughly divided to uncertainty-based sampling and representative sampling (Settles, 2009). Uncertainty-based sampling select samples that maximally reduce the uncertainty of the model (Tur et al., 2005). This, however, requires a well-trained model with decent confidence score estimations in order to perform well. Therefore, in this paper, we opt for the representative-sampling where the selected training instances are expected to be dissimilar to each other and representative enough to cover all important patterns in the whole data distribution (Agarwal et al., 2005; Wei et al., 2015). This naturally matches the objectives of k-means clustering which minimizes the within-cluster variances while maximizing the between-cluster variances to encourage the diversity and representativeness of each cluster (Krishna and Murty, 1999; Kanungo et al., 2002). As has been shown in image classification tasks, data points closer to the cluster centroids are usually most important, while other faraway points can even be safely removed without hurting model performance (Kaushal et al., 2018; Birodkar et al., 2019). Inspired by this, we propose a simple selection strategy which first clusters the whole unlabeled dataset with the K-means algorithm, and then from each cluster, selects the data point that is closest to the cluster centroid.

We conduct experiments on three popular text generation tasks: data-to-text, document summarization and question generation. The proposed selection strategy consistently outperforms random sampling and exhibits much smaller variance.

Contribution. We present a preliminary study on training instance selection for few-shot text generation and propose a selection strategy based on K-means clustering. The proposed method shows consistent superior performance over random sampling, which can be used to make most use of the annotation budget in practical applications. Meanwhile, the selected training instances can serve as a better benchmark for few-shot text generation since they are not biased towards specific generative methods and do not have the large variance issue as found in random sampling. We further perform a set of ablation studies to analyze what contributes to a good selection. Our findings can also benefit research in active learning (Konyushkova et al., 2017) since identifying the most informative training instances is a critical step before collecting more annotations through active learning.

2 Problem Formulation

Following the training scenario shown in Figure 1, we denote the unlabeled data as $U_1, U_2, \ldots, U_n$ where n is the data size. Depending on the downstream task, "data" can mean unlabeled structured data, documents and paragraphs respectively in the context of data-to-text, document summarization and question generation. We will select K instances from the whole unlabeled dataset, annotate them with reference text, and then train a neural generative model based on the annotated data. K is defined based on the annotation budget. In this work, since we focus on the few-shot scenario, K is set to be small (≤ 100). The goal is to *find the most representative K instances that can lead to the optimal performance when trained on them*.

3 Selection by K-means Clustering

The general idea of our proposed method is to first split the whole unlabeled data into K clusters, then select one data point from each cluster. Specifically, we first map each data point into a vector, then cluster the vectors with the K-means algorithm. The objective is sum of the squared errors (SSE), which is also called cluster inertia:

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{K} w_{i,j} ||x^i - \mu^j||_2^2 \qquad (1)$$

where μ^j is the centroid of the jth cluster. x^i is the embedding vector of U_i. $w_{i,j} = 1$ if x^i belongs to the cluster j and 0 otherwise. We optimize the objective function with the EM algorithm (Dempster et al., 1977) which iteratively assigns each data point into its closest cluster centroid. The initial centroid points are chosen based on the K-means++ algorithm (Arthur and Vassilvitskii, 2007). The first cluster center is chosen uniformly at random from the data points, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing cluster

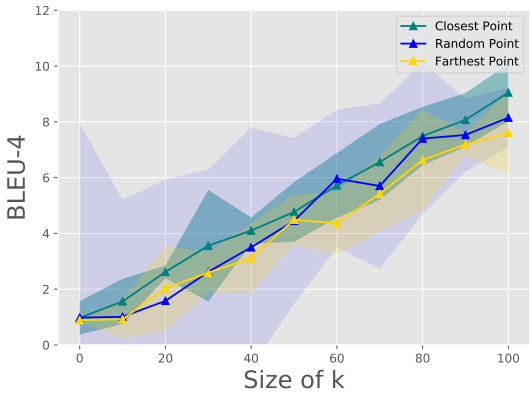


Figure 2: Ablation studies on the SQUAD corpus. Performance in BLEU-4 with increasing K between different variants of K-means where selection is based on the **closest point**, **Random point**, or **Farthest point** from the centroid.

center. By this means, we maximize the chance of spreading out the K initial cluster centers. We use 10 random seeds for selecting initial centers and the clustering with the minimum SSE is chosen.

After splitting them into K clusters, we pick from each cluster the data point that is closest to the center. We use the Euclidean distance to select, the same as the metric used for K-means clustering. The intuition is that the test performance usually depends on the nearest neighbor in the training set (Khandelwal et al., 2020; Rajani et al., 2020). Ideally data points closest to the cluster centers are most representative samples, selecting them will maximize the chance that a similar sample will be found in the training dataset.

4 Experiments

We perform our experiments on the following three representative datasets which cover three different text generation tasks:

1. Data-to-text: We use the dataset for the E2E challenge (Novikova et al., 2017) which contain 50,602 data-text pairs with 8 unique slots in the restaurant domain.

2. Document Summarization: We use the CNN/Dailymail dataset (non-anonymized version) (Hermann et al., 2015) which contains 312,084 document-summary pairs.

3. Question generation: We use the SQuAD dataset (Rajpurkar et al., 2016) with over 100k questions. Following Du et al. (2017), we focus on the answer-independent scenario to directly generate questions from passages.

For all experiments, we finetune the open-sourced Bart model (Lewis et al., 2020) as our generative model. Bart is pretrained with a denoising autoencoder objective on large amount of text data and has been the state-of-the-arts for many text generation tasks. To extract vectors used for clustering, we finetune the Bart model with its original self-supervised objective on the unlabeled data, then apply mean pooling over the last hidden states of the encoder.

In the later sections, we will first compare the model performance based on our proposed selection strategy and random sampling, then analyze the variance of them. Finally, we perform an ablation study to see the effects of in-cluster selection and embedding choices.

Comparison of Selection Strategies. In Table 1, we compare the model performance based on different selection strategies. Apart from random sampling and our proposed method, we also compare with a lower bound where all instances are randomly sampled from one cluster (within-cluster random). Adding this for comparison aims to illustrate that it is important to select diverse samples across different clusters. The performance scores are averaged over 10 different trials for each selection strategy. As can be seen, K-means based selections consistently outperforms the others. Within-cluster random sampling performs the worst, proving the importance of having diverse samples in the training instance. However, it is worth noting that although random sampling underperforms K-means selection on average, *its upper bound is much higher, suggesting the proposed K-means selection is by no means optimal*. There is still much room for improvement.

Variance of Model Performance. Table 1 also shows the variance of model performance with different selection strategies. The variance is computed based on 10 different runs. For within-cluster random sampling, the variance comes from both the choice of the cluster and the in-cluster sampling. For K-means selection, the variance comes from the choice of initial center points. We can see random sampling and within-cluster random sampling have a very large variance of up to 7.12 for $K = 100$. This further suggests that comparing few-shot models based on random sampling can be be prone to variability and prevent drawing reliable conclusions. K-means-based selection, on

	E2E			CNNDM			SQUAD		
	10	50	100	10	50	100	10	50	100
Random	4.38±7.12	11.57±4.29	26.22±2.58	13.51±6.47	24.81±3.77	35.24±2.89	1.23±6.22	3.33±5.89	7.65±3.61
IC-Random	2.15±4.58	9.80±2.62	24.71±2.71	12.30±3.89	24.71±2.45	33.29±1.92	1.34±3.23	1.79±3.77	6.97±2.55
K-means	**6.22**±2.33	**11.89**±1.39	**27.13**±2.22	**14.28±2.35**	**25.19**±3.28	**36.31**±1.08	**1.56**±2.34	**4.77**±3.61	**9.33**±2.15

Table 1: Comparisons of random sampling, within-cluster random sampling (IC-Random) and K-means selection on the E2E, CNNDM, and SQUAD corpus (BLEU-4 reported).

Embedding	E2E		CNNDM		SQUAD	
	Mean	Sum	Mean	Sum	Mean	Sum
BART	26.28	25.59	34.30	**34.46**	8.89	8.56
BART-FT	26.46	**26.32**	**36.31**	34.18	**9.55**	8.12
GloVe	25.18	23.36	33.59	31.45	7.99	7.56
FastText	**27.13**	24.85	33.23	34.30	9.33	**9.42**

Table 2: Finetuned BART generation performance comparison on **E2E**, **CNNDM**, and **SQUAD** for various embedding options for the *k-means selection* with k=100.

the contrary, is rather robust with random seeds. Therefore, for future work on few-shot text generation, we suggest that models be tested on instances selected from our proposed strategy for a fair comparison.

Effects of In-cluster Selection. In Figure 2, we show the effects of the in-cluster selection method. In our proposed method, within each cluster, we select one data point that is closest to the cluster center. To see whether it is important to select the closest data point, we compare with two selection variants that within each cluster, we select (1) one data point randomly sampled from the cluster, and (2) one data point that is farthest to the cluster center. We can observe that the choice of selection does have a big impact on the model performance. Choosing data points farthest to the cluster centers leads to the worst performance. This is consistent with previous findings (Kaushal et al., 2018; Birodkar et al., 2019) that data points farthest from cluster centers are usually outliers and less representative. Selecting them might mislead the model to capture non-generic patterns and thereby generalize poorly. In contrast, choosing data points closest to cluster centers performs slightly better than random selection. However, random selection has a much larger variance compared with closest/farthest point selection (shown as shadow).

Effects of Embedding Methods. As the K-means clustering is performed on top of the embedding vectors of unlabeled data, the choice of embedding methods could affect the performance on selected points. In Table 2, we show the effects of the different embedding methods. Apart from the finetuned Bart, we compare with embeddings extracted from (1) Bart without being finetuned on the task-specific data, (2) Glove (Pennington et al., 2014) and (3) FastText (Bojanowski et al., 2017), both finetuned on the task-specific data. For each embedding method, we compare using mean pooling and sum pooling to extract the final vector representation. The results show that finetuned Bart generally outperforms the other embedding choices. We attribute this to the similarity in the embedding space between selection with BART embeddings and and the BART generation model. Moreover, *FastText* offers a strong baseline as it does relatively well on two scenarios in E2E and SQUAD respectively. Further, we observe that *mean* pooling is generally better than the *sum* of word vectors, which is also observed in Chen et al. (2018).

Human Evaluation. To obtain further insights with respect to the generation outputs, five annotators were instructed to evaluate 100 samples for each of the three tasks to judge (1) whether the text is *fluent* (score 0-5 with 5 being fully fluent), and (2) whether it contains relevant information about its input source (*adequacy*). These scores are averaged and presented in Table 3. For **Random** selection, we sampled 10 outputs from each of the 10 trials to make it 100 samples, and the same goes for **IC-random**. We observe that the K-means algorithm select better subsets of the training samples that allow for better generalizability to unseen input sources. In particular, the outputs are generally more *adequate*. However, we see that the *fluency* of outputs remain relatively similar.

5 Conclusion

In this work, we target at the unexplored problem of training instance selection for few-shot text generation. We show that random sampling can lead to large variance and suboptimal performance. To address this problem, we propose a selection strategy based on K-mean clustering and demonstrate

	E2E	CNNDM	SQUAD
Random	4.08/4.15	**4.55**/3.27	**4.62**/3.84
IC-Random	4.32/3.54	3.62/3.01	4.23/2.74
K-means	**4.12/4.24**	4.32/**3.66**	4.51/**3.98**

Table 3: Human evaluation on 100 samples of the finetuned BART generation performance comparison on **E2E**, **CNNDM**, and **SQUAD**. Scores are presented as (*fluency* / *adequacy*).

that it consistently outperforms random sampling, and has much lower variance. We further perform a set of ablation studies to analyze the effects of data size, embedding and selection methods, showing that this is still much room for improvement. Future work can consider other clustering methods.

Acknowledgements

This research was funded in part by the German Research Foundation (DFG) as part of SFB 248 "Foundations of Perspicuous Software Systems". We sincerely thank the anonymous reviewers for their insightful comments that helped us to improve this paper.

References

Pankaj K Agarwal, Sariel Har-Peled, Kasturi R Varadarajan, et al. 2005. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30.

David Arthur and Sergei Vassilvitskii. 2007. k-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Maria-Florina Balcan, Andrei Broder, and Tong Zhang. 2007. Margin based active learning. In *International Conference on Computational Learning Theory*, pages 35–50. Springer.

Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. 2019. Semantic redundancies in image-classification datasets: The 10% you don't need. *arXiv preprint arXiv:1901.11409*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ernie Chang, Jeriah Caplinger, Alex Marin, Xiaoyu Shen, and Vera Demberg. 2020. Dart: A lightweight quality-suggestive data-to-text annotation tool. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, pages 12–17.

Ernie Chang, Vera Demberg, and Alex Marin. 2021a. Jointly improving language understanding and generation with quality-weighted weak supervision of automatic labeling. *Proceedings of EACL 2021*.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021b. Neural data-to-text generation with lm-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768.

Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1815–1826.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot nlg with pre-trained language model. *ACL*.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Association for Computational Linguistics (ACL)*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892.

Vishal Kaushal, Anurag Sahoo, Khoshrav Doctor, Narasimha Raju, Suyash Shetty, Pankaj Singh, Rishabh Iyer, and Ganesh Ramakrishnan. 2018. Learning from less data: Diversified subset selection and active learning in image classification tasks. *arXiv preprint arXiv:1805.11191*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. *ICLR*.

Ksenia Konyushkova, Sznitman Raphael, and Pascal Fua. 2017. Learning active learning from data. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4228–4238.

K Krishna and M Narasimha Murty. 1999. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. 2020. Explaining and improving model behavior with k nearest neighbor representations. *arXiv preprint arXiv:2010.09030*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Timo Schick and Hinrich Schütze. 2020a. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Burr Settles. 2009. Active learning literature survey.

Gokhan Tur, Dilek Hakkani-Tür, and Robert E Schapire. 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Coreference Resolution without Span Representations

Yuval Kirstain* **Ori Ram*** **Omer Levy**

Blavatnik School of Computer Science, Tel Aviv University
{yuval.kirstain,ori.ram,levyomer}@cs.tau.ac.il

Abstract

The introduction of pretrained language models has reduced many complex task-specific NLP models to simple lightweight layers. An exception to this trend is coreference resolution, where a sophisticated task-specific model is appended to a pretrained transformer encoder. While highly effective, the model has a very large memory footprint – primarily due to dynamically-constructed span and span-pair representations – which hinders the processing of complete documents and the ability to train on multiple instances in a single batch. We introduce a lightweight end-to-end coreference model that removes the dependency on span representations, handcrafted features, and heuristics. Our model performs competitively with the current standard model, while being simpler and more efficient.

1 Introduction

Until recently, the standard methodology in NLP was to design task-specific models, such as BiDAF for question answering (Seo et al., 2017) and ESIM for natural language inference (Chen et al., 2017). With the introduction of pretraining, many of these models were replaced with simple output layers, effectively fine-tuning the transformer layers below to perform the traditional model's function (Radford et al., 2018). A notable exception to this trend is *coreference resolution*, where a multi-layer task-specific model (Lee et al., 2017, 2018) is appended to a pretrained model (Joshi et al., 2019, 2020). This model uses intricate span and span-pair representations, a representation refinement mechanism, handcrafted features, pruning heuristics, and more. While the model is highly effective, it comes at a great cost in memory consumption, limiting the amount of examples that can be loaded on a large GPU to a single document, which often needs to be truncated or processed in sliding windows. Can this coreference model be simplified?

We present *start-to-end* (s2e) coreference resolution: a simple coreference model that does *not* construct span representations. Instead, our model propagates information to the span boundaries (i.e., its start and end tokens) and computes mention and antecedent scores through a series of bilinear functions over their contextualized representations. Our model has a significantly lighter memory footprint, allowing us to process multiple documents in a single batch, with no truncation or sliding windows. We do not use any handcrafted features, priors, or pruning heuristics.

Experiments show that our minimalist approach performs on par with the standard model, despite removing a significant amount of complexity, parameters, and heuristics. Without any hyperparameter tuning, our model achieves 80.3 F1 on the English OntoNotes dataset (Pradhan et al., 2012), with the best comparable baseline reaching 80.2 F1 (Joshi et al., 2020), while consuming less than a third of the memory. These results suggest that transformers can learn even difficult structured prediction tasks such as coreference resolution without investing in complex task-specific architectures.[1]

2 Background: Coreference Resolution

Coreference resolution is the task of clustering multiple mentions of the same entity within a given text. It is typically modeled by identifying entity mentions (contiguous spans of text), and predicting an *antecedent* mention a for each span q (query) that refers to a previously-mentioned entity, or a null-span ε otherwise.

Lee et al. (2017, 2018) introduce *coarse-to-fine* (c2f), an end-to-end model for coreference resolu-

*Equal contribution.

[1]Our code and model are publicly available: https://github.com/yuvalkirstain/s2e-coref

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 14–19
August 1–6, 2021. ©2021 Association for Computational Linguistics

tion that predicts, for each span q, an antecedent probability distribution over the candidate spans c:

$$P\left(a = c|q\right) = \frac{\exp(f(c,q))}{\sum_{c'} \exp(f(c',q))}$$

Here, $f(c,q)$ is a function that scores how likely c is to be an antecedent of q. This function is comprised of mention scores $f_m(c), f_m(q)$ (i.e. is the given span a mention?) and a separate antecedent score $f_a(c,q)$:

$$f(c,q) = \begin{cases} f_m(c) + f_m(q) + f_a(c,q) & c \neq \varepsilon \\ 0 & c = \varepsilon \end{cases}$$

Our model (Section 3) follows the scoring function above, but differs in how the different elements $f_m(\cdot)$ and $f_a(\cdot)$ are computed. We now describe how f_m and f_a are implemented in the c2f model.

Scoring Mentions In the c2f model, the mention score $f_m(q)$ is derived from a vector representation $\mathbf{v}_q$ of the span q (analogously, $f_m(c)$ is computed from $\mathbf{v}_c$). Let $\mathbf{x}_i$ be the contextualized representation of the i-th token produced by the underlying encoder. Every span representation is a concatenation of four elements: the representations of the span's start and end tokens $\mathbf{x}_{q_s}, \mathbf{x}_{q_e}$, a weighted average of the span's tokens $\hat{\mathbf{x}}_q$ computed via self-attentive pooling, and a feature vector $\phi(q)$ that represents the span's length:

$$\mathbf{v}_q = [\mathbf{x}_{q_s}; \mathbf{x}_{q_e}; \hat{\mathbf{x}}_q; \phi(q)]$$

The mention score $f_m(q)$ is then computed from the span representation $\mathbf{v}_q$:

$$f_m(q) = \mathbf{v}_m \cdot \text{ReLU}(\mathbf{W}_m \mathbf{v}_q)$$

where $\mathbf{W}_m$ and $\mathbf{v}_m$ are learned parameters. Then, span representations are enhanced with more global information through a refinement process that interpolates each span representation with a weighted average of its candidate antecedents. More recently, Xu and Choi (2020) demonstrated that this span refinement technique, as well as other modifications to it (e.g. entity equalization (Kantor and Globerson, 2019)) do not improve performance.

Scoring Antecedents The antecedent score $f_a(c,q)$ is derived from a vector representation of the span *pair* $\mathbf{v}_{(c,q)}$. This, in turn, is a function of the individual span representations $\mathbf{v}_c$ and $\mathbf{v}_q$, as well as a vector of handcrafted features $\phi(c,q)$ such as the distance between the spans c and q, the document's genre, and whether c and q were said/written by the same speaker:

$$\mathbf{v}_{(c,q)} = [\mathbf{v}_c; \mathbf{v}_q; \mathbf{v}_c \circ \mathbf{v}_q; \phi(c,q)]$$

The antecedent score $f_a(c,q)$ is parameterized with $\mathbf{W}_a$ and $\mathbf{v}_a$ as follows:

$$f_a(c,q) = \mathbf{v}_a \cdot \text{ReLU}(\mathbf{W}_a \mathbf{v}_{(c,q)})$$

Pruning Holding the vector representation of every possible span in memory has a space complexity of $O(n^2 d)$ (where n is the number of input tokens, and d is the model's hidden dimension). This problem becomes even more acute when considering the space of span *pairs* ($O(n^4 d)$). Since this is not feasible, candidate mentions and antecedents are pruned through a variety of model-based and heuristic methods.

Specifically, mention spans are limited to a certain maximum length ℓ. The remaining mentions are then ranked according to their scores $f_m(\cdot)$, and only the top λn are retained, while avoiding overlapping spans. Antecedents (span pairs) are further pruned using a lightweight antecedent scoring function (which is added to the overall antecedent score), retaining only a constant number of antecedent candidates c for each target mention q.

Training For each remaining span q, the training objective optimizes the marginal log-likelihood of all of its unpruned gold antecedents c, as there may be multiple mentions referring to the same entity:

$$\log \sum_{c} P(a = c|q)$$

Processing Long Documents Due to the c2f model's high memory consumption and the limited sequence length of most pretrained transformers, documents are often split into segments of a few hundred tokens each (Joshi et al., 2019). Recent work on efficient transformers (Beltagy et al., 2020) has been able to shift towards processing complete documents, albeit with a smaller model (base) and only one training example per batch.

3 Model

We present *start-to-end* (s2e) coreference resolution, a simpler and more efficient model with respect to c2f (Section 2). Our model utilizes the endpoints of a span (rather than all span tokens) to compute the mention and antecedent scores $f_m(\cdot)$

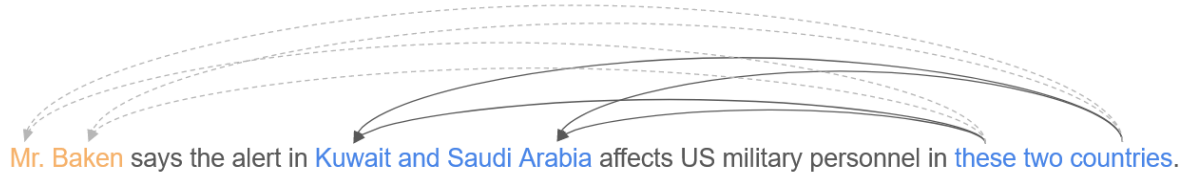


Figure 1: The antecedent score $f_a(c, q)$ of a query mention $q = (q_s, q_e)$ and a candidate antecedent $c = (c_s, c_e)$ is defined via bilinear functions over the representations of their endpoints c_s, c_e, q_s, q_e. Solid lines reflect factors participating in positive examples (coreferring mentions), and dashed lines correspond to negative examples.

and $f_a(\cdot, \cdot)$ *without* constructing span or span-pair representations; instead, we rely on a combination of lightweight bilinear functions between pairs of endpoint token representations. Furthermore, our model does not use any handcrafted features, does not prune antecedents, and prunes mention candidates solely based on their mention score $f_m(q)$.

Our computation begins by extracting a *start* and *end* token representation from the contextualized representation $\mathbf{x}$ of each token in the sequence:

$$\mathbf{m}^s = \text{GeLU}(\mathbf{W}_m^s \mathbf{x}) \qquad \mathbf{m}^e = \text{GeLU}(\mathbf{W}_m^e \mathbf{x})$$

We then compute each mention score as a biaffine product over the start and end tokens' representations, similar to Dozat and Manning (2017):

$$f_m(q) = \mathbf{v}_s \cdot \mathbf{m}_{q_s}^s + \mathbf{v}_e \cdot \mathbf{m}_{q_e}^e + \mathbf{m}_{q_s}^s \cdot \mathbf{B}_m \cdot \mathbf{m}_{q_e}^e$$

The first two factors measure how likely the span's start/end token q_s/q_e is a beginning/ending of an entity mention. The third measures whether those tokens are the boundary points of the *same* entity mention. The vectors $\mathbf{v}_s, \mathbf{v}_e$ and the matrix $\mathbf{B}_m$ are the trainable parameters of our mention scoring function f_m. We efficiently compute mention scores for all possible spans while masking spans that exceed a certain length ℓ.[2] We then retain only the top-scoring λn mention candidates to avoid $O(n^4)$ complexity when computing antecedents.

Similarly, we extract *start* and *end* token representations for the antecedent scoring function f_a:

$$\mathbf{a}^s = \text{GeLU}(\mathbf{W}_a^s \mathbf{x}) \qquad \mathbf{a}^e = \text{GeLU}(\mathbf{W}_a^e \mathbf{x})$$

Then, we sum over four bilinear functions:

$$\begin{aligned} f_a(c, q) = \mathbf{a}_{c_s}^s \cdot \mathbf{B}_a^{ss} \cdot \mathbf{a}_{q_s}^s + \mathbf{a}_{c_s}^s \cdot \mathbf{B}_a^{se} \cdot \mathbf{a}_{q_e}^e \\ + \mathbf{a}_{c_e}^e \cdot \mathbf{B}_a^{es} \cdot \mathbf{a}_{q_s}^s + \mathbf{a}_{c_e}^e \cdot \mathbf{B}_a^{ee} \cdot \mathbf{a}_{q_e}^e \end{aligned}$$

Each component measures the compatibility of the spans c and q by an interaction between different boundary tokens of each span. The first component compares the *start* representations of c and q, while the fourth component compares the *end* representations. The second and third facilitate a cross-comparison of the *start* token of span c with the *end* token of span q, and vice versa. Figure 1 (bottom) illustrates these interactions.

This calculation is equivalent to computing a bilinear transformation between the concatenation of each span's boundary tokens' representations:

$$f_a(c, q) = [\mathbf{a}_{c_s}^s; \mathbf{a}_{c_e}^e] \cdot \mathbf{B}_a \cdot [\mathbf{a}_{q_s}^s; \mathbf{a}_{q_e}^e]$$

However, computing the factors *directly* bypasses the need to create n^2 explicit span representations. Thus, we avoid a theoretical space complexity of $O(n^2 d)$, while keeping it equivalent to that of a transformer layer, namely $O(n^2 + nd)$.

4 Experiments

Dataset We train and evaluate on two datasets: the document-level English OntoNotes 5.0 dataset (Pradhan et al., 2012), and the GAP coreference dataset (Webster et al., 2018). The OntoNotes dataset contains speaker metadata, which the baselines use through a hand-crafted feature that indicates whether two spans were uttered by the same speaker. Instead, we insert the speaker's name to the text every time the speaker changes, making the metadata available to any model.

Pretrained Model We use Longformer-Large (Beltagy et al., 2020) as our underlying pretrained model, since it is able to process long documents without resorting to sliding windows or truncation.

Baseline We consider Joshi et al.'s (2019) expansion to the c2f model as our baseline. Specifically, we use the implementation of Xu and Choi (2020) with minor adaptations for supporting Longformer. We do not use higher-order inference, as Xu and Choi (2020) demonstrate that it does not result in significant improvements. We train the baseline

[2] While pruning by length is not necessary for efficiency, we found it to be a good inductive bias.

Model	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
c2f + SpanBERT-Large	85.7	**85.3**	85.5	79.5	**78.7**	**79.1**	**76.8**	75.0	75.9	80.2
c2f + Longformer-Base	85.0	85.0	85.0	77.8	77.8	77.8	75.6	74.2	74.9	79.2
c2f + Longformer-Large	86.0	83.2	84.6	78.9	75.5	77.2	76.7	68.7	72.5	78.1
s2e + Longformer-Large	**86.5**	85.1	**85.8**	**80.3**	77.9	**79.1**	**76.8**	**75.4**	**76.1**	**80.3**

Table 1: Performance on the test set of the English OntoNotes 5.0 dataset. *c2f* refers to the course-to-fine approach of Lee et al. (2017, 2018), as ported to pretrained transformers by Joshi et al. (2019).

	Masc	**Fem**	**Bias**	**Overall**
c2f + SpanBERT-Large	90.5	**86.3**	**0.95**	**88.4**
c2f + Longformer-Base	87.6	82.3	0.94	84.9
c2f + Longformer-Large	90.1	85.4	**0.95**	87.8
s2e + Longformer-Large	**90.6**	85.8	**0.95**	88.3

Table 2: Performance on the test set of the GAP coreference dataset. The reported metrics are F1 scores.

Model	Memory (GB)
c2f + SpanBERT-Large	16.2
c2f + Longformer-Base	12.0
c2f + Longformer-Large	15.7
s2e + Longformer-Large	**4.3**

Table 3: Peak GPU memory usage during inference on OntoNotes, when processing one document at a time.

model over three pretrained models: Longformer-Base, Longformer-Large, and SpanBERT-Large (Beltagy et al., 2020; Joshi et al., 2020).

Hyperparameters All models use the same hyperparameters as the baseline. The only hyperparameters we change are the maximum sequence length and batch size, which we enlarge to fit as many tokens as possible into a 32GB GPU.[3] For our model, we use dynamic batching with 5,000 max tokens, which allows us to fit an average of 5-6 documents in every training batch. The baseline, however, has a much higher memory footprint, and is barely able to fit a single example with Longformer-Base (max 4,096 tokens). When combining the baseline with SpanBERT-Large or Longformer-Large, the baseline must resort to sliding windows to process the full document (512 and 2,048 tokens, respectively).

Performance Table 1 and Table 2 show that, despite our model's simplicity, it performs as well as the best performing baseline. Our model with Longformer-Large achieves 80.3 F1 on OntoNotes, while the best performing baseline achieves 80.2 F1. When the baseline model is combined with either version of Longformer, it is not able to reach the same performance level as our model. We see similar trends for GAP. Our findings indicate that there is little to lose from simplifying the coreference resolution architecture, while there are potential gains to be had from optimizing with larger batches.

Efficiency We also compare our model's memory usage using the OntoNotes development set. Table 3 shows that our implementation is at least three times more memory efficient than the baseline. This improvement results from a combination of three factors: (1) the fact that our model is lighter on memory and does not need to construct span or span-pair representations, (2) our simplified framework, which does not use sliding windows, and (3) our implementation, which was written "from scratch", and might thus be more (or less) efficient than the original.

5 Related Work

Recent work on memory-efficient coreference resolution sacrifices speed and parallelism for guarantees on memory consumption. Xia et al. (2020) and Toshniwal et al. (2020) present variants of the c2f model (Lee et al., 2017, 2018) that use an iterative process to maintain a fixed number of span representations at all times. Specifically, spans are processed sequentially, either joining existing clusters or forming new ones, and an eviction mechanism ensures the use of a constant number of clusters. While these approach constrains the space complexity, their sequential nature slows down the computation, and slightly deteriorates the performance. Our approach is able to alleviate the large

[3] We made one exception, and tried to tune the Longformer-Large baseline's hyperparameters. Despite our efforts, it still performs worse than Longformer-Base.

memory footprint of c2f while maintaining fast parallel processing and high performance.

CorefQA (Wu et al., 2020) propose an alternative solution by casting the task of coreference resolution as one of extractive question answering. It first detects potential mentions, and then creates dedicated queries for each one, creating a pseudo-question-answering instance for each candidate mention. This method significantly improves performance, but at the cost of processing hundreds of individual context-question-answer instances for a single document, substantially increasing execution time. Our work provides a simple alternative, which can scale well in terms of both speed and memory.

6 Conclusion

We introduce a new model for coreference resolution, suggesting a lightweight alternative to the sophisticated model that has dominated the task over the past few years. Our model is competitive with the baseline, while being simpler and more efficient. This finding once again demonstrates the spectacular ability of deep pretrained transformers to model complex natural language phenomena.

Acknowledgements

This research was funded by the Blavatnik Fund, the Alon Scholarship, Intel Corporation, and the Yandex Initiative in Machine Learning.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR 2017*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8519–8526, Online. Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624, Online. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Enhancing Entity Boundary Detection for Better Chinese Named Entity Recognition

Chun Chen, Fang Kong*
School of Computer Science and Technology
Soochow University
20195227037@stu.suda.edu.cn, kongfang@suda.edu.cn

Abstract

In comparison with English, due to the lack of explicit word boundary and tenses information, Chinese Named Entity Recognition (NER) is much more challenging. In this paper, we propose a boundary enhanced approach for better Chinese NER. In particular, our approach enhances the boundary information from two perspectives. On one hand, we enhance the representation of the internal dependency of phrases by an additional Graph Attention Network(GAT) layer. On the other hand, taking the entity head-tail prediction (i.e., boundaries) as an auxiliary task, we propose an unified framework to learn the boundary information and recognize the NE jointly. Experiments on both the OntoNotes and the Weibo corpora show the effectiveness of our approach.

1 Introduction

Given a sentence, the NER task aims to identify the noun phrases having special meanings that predefined. Due to its importance on many downstream tasks, such as relation extraction(Ji et al., 2017), coreference resolution(Clark and Manning, 2016) and knowledge graphs(Zhang et al., 2019), NER has attracted much attention for long time.

In comparison with English, due to the lack of explicit word boundary and tenses information, Chinese NER is much more challenging. In fact, the performance of the current SOTAs in Chinese is far inferior to that in English, the gap is about 10% in F1-measure. In this paper, we propose a boundary enhancing approach for better Chinese NER.

Firstly, using Star-Transformer(Guo et al., 2019), we construct a lightweight baseline system. Benefit from the unique star topological structure, Star-Transformer is more dominant in representing long distance sequence, and thus, our baseline achieves comparable performance to the SOTAs. Considering the deficiency in the representation of local sequence information, we then try to enhance the local boundary information. In particular, our approach enhances the boundary information from two perspectives. On one hand, we add an additional GAT(Veličković et al., 2017) layer to capture the internal dependency of phrases. In this way, boundaries can be distinguished implicitly, while the semantic information within the phrase is enhanced. On the other hand, we add an auxiliary task to predict the head and tail of entities. In this way, using the framework of multi-tasking learning, we can learn the boundary information explicitly and help the NER task. Experiments show the effectiveness of our approach. It should be noted that, our approach obtains the new state-of-the-art results on both the OntoNotes and the Weibo corpora. That means our approach can perform well for both written and non-written texts.

2 Related Work

As is well known, most researches cast the NER task as a traditional sequence labelling problem, and many models extending the Bi-LSTM+CRF architecture are proposed (Huang et al., 2015; Chiu and Nichols, 2016; Dong et al., 2016; Lample et al., 2016; Ma and Hovy, 2016). Although the attention-based model, i.e., Transformer(Vaswani et al., 2017), has gradually surpassed the traditional RNN model(Zaremba et al., 2014) in various fields, Yan et al. (2019) has verified that the fully connected Transformer mechanism does not work well on NER. Until recently, some researches show that Star-Transformer can work well on NER owing to its lightweight topological structure(Guo et al., 2019; Chen et al., 2020). Moreover, lexical and dependent information has been widely used in this task (Zhang and Yang, 2018; Ma et al., 2020; Li et al., 2020; Gui et al., 2019; Sui et al., 2019; Tang et al., 2020) to better capture local semantic information.

In this paper, using Star-transformer as our base-

*Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 20–25
August 1–6, 2021. ©2021 Association for Computational Linguistics

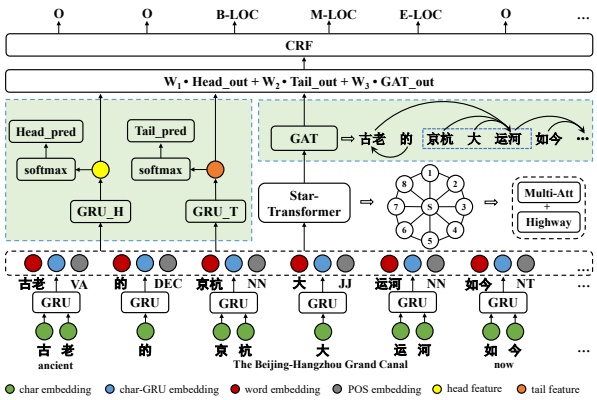


Figure 1: The general architecture for the boundary enhanced model.

line, we mainly focus on enhancing the boundary information to improve Chinese NER.

3 Model

We also treat NER as a sequence labeling task, decoding with a classical CRF(Lafferty et al., 2001). Figure 1 shows the complete model. We can find that the encoder of our model consists of three parts, i.e., GRU-based head and tail representation layer, Star-transformer based contextual embedding layer, and GAT-based dependency embedding layer.

3.1 Token embedding layer

Considering the lack of explit word boundary, we combine word-level represention with character, avoiding the error propagation caused by word segmentation.

For a given sentence, we represent each word and character by looking up the pre-trained word embeddings[1](Li et al., 2018). The sequence of character embeddings contained in a word will be fed to a bi-direction GRU layer. The hidden state of bi-direction GRU can be expressed as folowing:

$$\overrightarrow{h}_i^t = \overrightarrow{GRU}(x_i^t, \overrightarrow{h}_{i-1}^t) \quad (1)$$

$$\overleftarrow{h}_i^t = \overleftarrow{GRU}(x_i^t, \overleftarrow{h}_{i+1}^t) \quad (2)$$

$$h_i^t = [\overrightarrow{h}_i^t; \overleftarrow{h}_i^t] \quad (3)$$

where x_i^t is the token representation, $\overrightarrow{h}_i^t$ and $\overleftarrow{h}_i^t$ denote the t-th forward and backward hidden state of GRU layer.

The final token representation is obtained as equation(4) $\sim$ (6):

$$x_i^w = e(word_i) \quad (4)$$

$$x_i^c = GRU(e(char_i)) \quad (5)$$

$$x_i = [x_i^w; x_i^c; pos_i] \quad (6)$$

where [;] denotes concatenation, and pos_i is the Part-of-Speech tagging of $word_i$.

3.2 Star-transformer based contextual embedding layer

Star-Transformer abandons redundant connections and has an approximate ability to model the long-range dependencies. For NER task, entities are sparse, so it is unnecessary to pay attention on all nodes in the sentence all the time. We utilize this structured model to encode the words in a sentence, which shows comparable performance with the traditional RNN models, but with the capability of capturing long-range dependencies.

3.2.1 Multi-Head Attention

Transformer employs h attention heads to implement self-attention on an input sequence separately. The result of each attention head will be integrated together, called Multi-Head Attention.

Given a sequence of vectors X, we use a query vector Q to soft select the relevant information with attention:

$$Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V \quad (7)$$

$$K = XW^K, V = XW^V \quad (8)$$

where W^K and W^V are learnable parameters. Then Multi-Head Attention can be defined as equation(9) $\sim$ (10):

$$MulAtt = (z_1 \oplus z_2 \oplus \cdots \oplus z_h) \cdot W^o \quad (9)$$

$$z_i = Att(QW_i^Q, KW_i^K, VW_i^V) \quad (10)$$

where $\oplus$ denotes concatenation, and W^o, W_i^Q, W_i^K, W_i^V are learnable parameters.

3.2.2 Star-Transformer Encoder

The topological structure of Star-Transformer is made up of one relay node and n satellite nodes. The state of i-th satellite node represents the feature of the i-th token in a text sequence. The relay node acts as a virtual hub to gather and scatter information from and to all the satellite nodes(Guo et al., 2019).

[1]https://github.com/Embedding/Chinese-Word-Vectors

Star-Transformer proposes a time-step cyclic updating method, in which each satellite node is initialized by the input vector, and the relay node is initialized as the average value of all tokens. The status of each satellite node is updated according to its adjacent nodes, including the previous node in the previous round h_{i-1}^{t-1}, the current node in the previous round h_i^{t-1}, the next node in the previous round h_{i+1}^{t-1}, the current node e^i and the relay node in the previous round s^{t-1}. The update process is shown in the equation(11) $\sim$ (12):

$$C_i^t = [h_{i-1}^{t-1}; h_i^{t-1}; h_{i+1}^{t-1}; e^i; s^{t-1}] \quad (11)$$

$$h_i^t = MulAtt(h_i^{t-1}, C_i^t, C_i^t) \quad (12)$$

where C_i^t denotes contextual information of i-th.

The update of relay node is determined by the information of all the satellite nodes and the status of the previous round :

$$s^t = MulAtt(s^{t-1}, [s^{t-1}; H^t], [s^{t-1}; H^t]) \quad (13)$$

3.2.3 Highway Networks

Highway Networks(Srivastava et al., 2015) can alleviate the blocked gradient backflow when the network deepens. Such gating mechanisms can be of vital significance to Transformer(Chai et al., 2020). We use Highway Networks to mitigate the depth and complexity of Star-Transformer.

After calculating the Multi-Head Attention, a new branch dominated by Highway Networks joins in, indicating the self-updating and dynamic adjustment of satellite node.

$$g = \sigma(w_1 h_i + b_1) \quad (14)$$

$$f(h_i) = w_2 h_i + b_2 \quad (15)$$

$$HW(h_i) = (1-g) \cdot h_i + g \cdot f(h_i) \quad (16)$$

where w_1, w_2, b_1, b_2 are learnable parameters, and σ is the activation function.

Finally, the updated satellite node is denoted as:

$$h_i = HW(h_i) + MulAtt(h_i, C_i, C_i) \quad (17)$$

Highway Networks not only enhances the inherent characteristics of the satellite nodes, but also avoids gradient blocking.

3.3 GAT-based dependency embedding layer

In this work, we propose the use of dependencies between words to construct graph neural networks. The dependency is directional, and the current word is only related to the word with shared edge. This kind of directed linkage further obtains the internal structural information of the entity, enriching the sequential representation.

Graph Attention Networks(GAT)(Veličković et al., 2017), leveraging masked self-attention layers to assign different importance to neighbouring nodes, works well with our work.

The attention coefficient e_{ij} and α_{ij} represents the importance of node j to node i:

$$e_{ij} = att(W\overrightarrow{h}_i, W\overrightarrow{h}_j) \quad (18)$$

$$\alpha_{ij} = softmax_j(e_{ij}) \quad (19)$$

$$= \frac{exp(e_{ij})}{\sum_{k \epsilon N_i} exp(e_{ik})} \quad (20)$$

$$= \frac{exp(LeakyReLU(\overrightarrow{a}^T[Wh_i \oplus Wh_j]))}{\Sigma_{k \epsilon N_i} exp(LeakyReLU(\overrightarrow{a}^T[Wh_i \oplus Wh_k]))} \quad (21)$$

A GAT operation with K independent attention heads can be expressed as:

$$\overrightarrow{h}'_i = \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \epsilon N_i} \alpha_{ij}^k W^k \overrightarrow{h}_j) \quad (22)$$

where $\oplus$ denotes concatenation, W and $\overrightarrow{a}$ are learnable parameters, N_i is the neighborhood of node i, σ is the activation function.

In addition to the strong focus on the associated nodes of GAT layer, it can well make up for the deficiency of Star-Transformer in capturing the internal dependency of the phrases.

3.4 GRU-based head and tail representation layer

While GAT is effective in capturing internal dependency within an entity, the boundary of the entity need to be strengthened. We then regard the entity boundary detection as binary classification task, which trains with NER at the same time, giving NER clear entity boundary information.

During training phase, two separate GRU layers are used to make head and tail prediction of the entities, whose hidden features are added with the output of GAT layer:

$$H_h = GRU_{head}(x_i) \quad (23)$$

$$H_t = GRU_{tail}(x_i) \quad (24)$$

$$H = W_1 \cdot H_h + W_2 \cdot H_t + W_3 \cdot H_{GAT} \quad (25)$$

W_1, W_2, W_3 are learnable parameters, and H is the final input for CRF.

	OntoNotes						Weibo						
	OntoNotes V4.0			OntoNotes V5.0			Named Entity			Nominal Mention			Overall
Models	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	F1(%)
Zhang and Yang (2018)	76.35	71.56	73.88	-	-	-	-	-	53.04	-	-	62.25	58.79
Ma et al. (2020)	77.31	73.85	75.54	-	-	-	-	-	56.99	-	-	61.41	61.24
Li et al. (2020)	-	-	76.45	-	-	-	-	-	-	-	-	-	63.42
Jie and Lu (2019)	-	-	-	77.40	77.41	77.40	-	-	-	-	-	-	-
Gui et al. (2019)	76.13	73.68	74.89	-	-	-	-	-	55.34	-	-	64.98	60.21
Sui et al. (2019)	75.06	74.52	74.79	-	-	-	67.31	48.61	56.45	75.15	62.63	68.32	63.09
Tang et al. (2020)	76.59	75.17	75.87	-	-	-	-	-	59.08	-	-	68.61	63.63
Star(baseline)	73.40	76.50	74.92	75.41	75.66	75.53	78.67	55.92	65.37	88.16	**69.07**	77.46	68.15
Star + GAT	77.33	76.03	76.67	77.03	79.90	78.44	77.30	59.72	67.38	**90.85**	66.49	76.79	68.34
Star + MultiTask	78.64	**80.78**	79.69	77.60	80.01	78.79	**80.39**	58.29	67.58	89.86	68.56	**77.78**	68.61
Star + GAT + MultiTask	**79.25**	80.66	**79.95**	**78.22**	**80.88**	**79.53**	78.92	**62.09**	**69.50**	88.67	68.56	77.33	**70.14**

Table 1: Performance on OntoNotes V4.0, OntoNotes V5.0 and Weibo. Named Entity is the same to the entity of OntoNotes, while Nominal Mention is the reference words which have the property of nouns.

3.5 Model Learning

Entities boundaries are not only the task we deal with, but the perfect natural assistance by NER, which transform from outside to inside of the mention and vice versa.

The multi-task loss function is composed of the categorical cross-entroy loss for boundary detection and entity categorical label prediction:

$$L_{multi} = L_{head} + L_{tail} + L_{label} \quad (26)$$

4 Experiments

4.1 Datasets

The label in our work is marked by BIESO, and we use Precision(P), Recall(R) and $F1$ score($F1$) as evaluation metrics.

OntoNotes V4.0[2](Pradhan, 2011) is a Chinese dataset and consists of texts from news domain. We use the same split as Zhang and Yang (2018).

OntoNotes V5.0[3](Pradhan et al., 2013) is also a Chinese dataset from news domain, but with larger scale and more entity types. We use the same split as Jie and Lu (2019).

Weibo NER[4](Peng and Dredze, 2015) contains annotated NER messages drawn from the social meida Sina Weibo. We use the same split as Peng and Dredze (2015).

Additionally, the tool used to parse syntactic dependency in this paper is DDParser[5].

4.2 Results and Analysis

We conduct experiments on the OntoNotes and Weibo corpora and compare the results with the existing models, as shown in table 1[6].

	OntoNotes V4.0			OntoNotes V5.0		
error types	TE	UE	BE	TE	UE	BE
Star	2236	1912	151	1921	1896	139
Star + GAT	1787	1916	140	1877	1596	169
Star + MultiTask	1772	**1563**	114	1814	1590	127
Star + GAT + MultiTask	**1701**	1564	**108**	**1762**	**1505**	**121**

Table 2: Entity recognition errors of our models, including Type Error(TE), Unidentification Error(UE) and Boundary Error(BE).

We begin by establishing a Star-Transformer baseline, which is more effective on the smaller social media Weibo corpus than OntoNotes. Star-Transformer could be superior to all existing models in Weibo, at least 6.29%(F1) and 8.85%(F1) for Named Entity(NE) and Nominal Entity(NM).

Considering the structural peculiarity of OntoNotes, where entities have similar composition, we utilize GAT to simulate the feature inside the entity. The precision on the OntoNotes are both improved by 3.93% and 1.62%. Futhermore, boundary prediction used as multi-task has been trained with label classification, supplying local sequence information for NER. Tabel 2 shows the number of different entity recognition errors of our models, including Type Error(TE), Unidentification Error(UE) and Boundary Error(BE).The addition of entity head-tail prediction reduces the number of boundary errors on OntoNotes V4.0 by 37. There is no doubt that the boundary enhanced model are quite profitable to the recognition of both entity boundary and entity type.

For Weibo, NE and NM illustrate different performance. The more standard NE has a similar performance to OntoNotes, while NM shows less

[2]https://catalog.ldc.upenn.edu/LDC2011T13
[3]https://catalog.ldc.upenn.edu/LDC2013T19
[4]https://github.com/cchen-nlp/weiboNER
[5]https://github.com/baidu/DDParser

[6]Our code is available at: https://github.com/cchen-reese/Boundary-Enhanced-NER.

impact from GAT, due to its short length and non-structue.

Combining the respective advantages of the three layers above, an unified and lightweight model can be applied to Chinese NER, getting the new state-of-the-art results on both the OntoNotes and Weibo corpora.

5 Conclusion

In this paper, we mainly focus on the impact of boundary information on Chinese NER. We firstly propose a Star-transformer based NER system. Then both explicit head and tail boundary information and Dependency GAT-based implicit boundary information are combined to improve Chinese NER. Experiments on both the OntoNotes and the Weibo corpora show the effectiveness of our approach.

Acknowledgement

This work was supported by the National Key R&D Program of China under Grant No. 2020AAA0108600, Project 61876118 under the National Natural Science Foundation of China and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Yekun Chai, Jin Shuo, and Xinwen Hou. 2020. Highway transformer: Self-gating enhanced self-attentive networks. *arXiv preprint arXiv:2004.08178*.

Chun Chen, Mingyang Li, and Fang Kong. 2020. Lightweight named entity recognition for weibo based on word and character. In *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, pages 402–413.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*, pages 239–250. Springer.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuan-Jing Huang. 2019. A lexicon-based graph neural network for chinese ner. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1039–1049.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. *arXiv preprint arXiv:1902.09113*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, et al. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, volume 3060.

Zhanming Jie and Wei Lu. 2019. Dependency-guided lstm-crf for named entity recognition. *arXiv preprint arXiv:1909.10148*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. *arXiv preprint arXiv:1805.06504*.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. Flat: Chinese ner using flat-lattice transformer. *arXiv preprint arXiv:2004.11795*.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuan-Jing Huang. 2020. Simplify the usage of lexicon in chinese ner. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

Sameer Pradhan. 2011. Proceedings of the fifteenth conference on computational natural language learning: Shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3821–3831.

Zhuo Tang, Boyan Wan, and Li Yang. 2020. Word-character graph convolution network for chinese named entity recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. Tener: Adapting transformer encoder for name entity recognition. *arXiv preprint arXiv:1911.04474*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 96–104.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*.

Difficulty-Aware Machine Translation Evaluation

Runzhe Zhan* **Xuebo Liu*** **Derek F. Wong†** **Lidia S. Chao**
NLP^{2}CT Lab, Department of Computer and Information Science, University of Macau
nlp2ct.{runzhe,xuebo}@gmail.com, {derekfw,lidiasc}@um.edu.mo

Abstract

The high-quality translation results produced by machine translation (MT) systems still pose a huge challenge for automatic evaluation. Current MT evaluation pays the same attention to each sentence component, while the questions of real-world examinations (e.g., university examinations) have different difficulties and weightings. In this paper, we propose a novel *difficulty-aware MT evaluation* metric, expanding the evaluation dimension by taking translation difficulty into consideration. A translation that fails to be predicted by most MT systems will be treated as a difficult one and assigned a large weight in the final score function, and conversely. Experimental results on the WMT19 English↔German Metrics shared tasks show that our proposed method outperforms commonly-used MT metrics in terms of human correlation. In particular, our proposed method performs well even when all the MT systems are very competitive, which is when most existing metrics fail to distinguish between them. The source code is freely available at https://github.com/NLP2CT/Difficulty-Aware-MT-Evaluation.

1 Introduction

The human labor needed to evaluate machine translation (MT) evaluation is expensive. To alleviate this, various automatic evaluation metrics are continuously being introduced to correlate with human judgements. Unfortunately, cutting-edge MT systems are too close in performance and generation style for such metrics to rank systems. Even for a metric whose correlation is reliable in most cases, empirical research has shown that it poorly correlates with human ratings when evaluating competitive systems (Ma et al., 2019; Mathur et al., 2020), limiting the development of MT systems.

Current MT evaluation still faces the challenge of how to better evaluate the overlap between the reference and the model hypothesis taking into consideration *adequacy* and *fluency*, where all the evaluation units are treated the same, i.e., all the matching scores have an equal weighting. However, in real-world examinations, the questions vary in their difficulty. Those questions which are easily answered by most subjects tend to have low weightings, while those which are hard to answer have high weightings. A subject who is able to solve the more difficult questions can receive a high final score and gain a better ranking. MT evaluation is also a kind of examination. For bridging the gap between human examination and MT evaluation, it is advisable to incorporate a *difficulty* dimension into the MT evaluation metric.

In this paper, we take translation difficulty into account in MT evaluation and test the effectiveness on a representative MT metric BERTScore (Zhang et al., 2020) to verify the feasibility. More specifically, the difficulty is first determined across the systems with the help of pairwise similarity, and then exploited as the weight in the final score function for distinguishing the contribution of different sub-units. Experimental results on the WMT19 English↔German evaluation task show that difficulty-aware BERTScore has a better correlation than do the existing metrics. Moreover, it agrees very well with the human rankings when evaluating competitive systems.

2 Related Work

The existing MT evaluation metrics can be categorized into the following types according to their underlying matching sub-units: n-gram based (Papineni et al., 2002; Doddington, 2002; Lin and Och, 2004; Han et al., 2012; Popović, 2015),

*Equal contribution
†Corresponding author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 26–32
August 1–6, 2021. ©2021 Association for Computational Linguistics

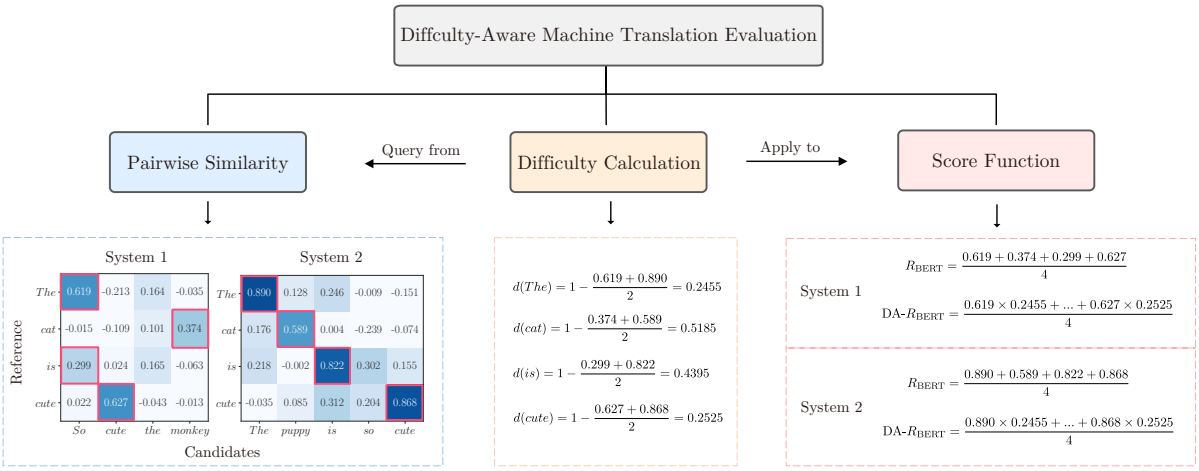


Figure 1: Illustration of combining difficulty weight with BERTScore. R_{BERT} denotes the vanilla recall-based BERTScore while DA-R_{BERT} denotes the score augmented with translation difficulty.

edit-distance based (Snover et al., 2006; Leusch et al., 2006), alignment-based (Banerjee and Lavie, 2005), embedding-based (Zhang et al., 2020; Chow et al., 2019; Lo, 2019) and end-to-end based (Sellam et al., 2020). BLEU (Papineni et al., 2002) is widely used as a vital criterion in the comparison of MT system performance but its reliability has been doubted on entering neural machine translation age (Shterionov et al., 2018; Mathur et al., 2020). Due to the fact that BLEU and its variants only assess surface linguistic features, some metrics leveraging contextual embedding and end-to-end training bring semantic information into the evaluation, which further improves the correlation with human judgement. Among them, BERTScore (Zhang et al., 2020) has achieved a remarkable performance across MT evaluation benchmarks balancing speed and correlation. In this paper, we choose BERTScore as our testbed.

3 Our Proposed Method

3.1 Motivation

In real-world examinations, the questions are empirically divided into various levels of difficulty. Since the difficulty varies from question to question, the corresponding role a question plays in the evaluation does also. Simple question, which can be answered by most of the subjects, usually receive of a low weighting. But a difficult question, which has more discriminative power, can only be answered by a small number of good subjects, and thus receives a higher weighting.

Motivated by this evaluation mechanism, we measure difficulty of a translation by viewing the MT systems and sub-units of the sentence as the subjects and questions, respectively. From this perspective, the impact of the sentence-level sub-units on the evaluation results supported a differentiation. Those sub-units that may be incorrectly translated by most systems (e.g., polysemy) should have a higher weight in the assessment, while easier-to-translate sub-units (e.g., the definite article) should receive less weight.

3.2 Difficulty-Aware BERTScore

In this part, we aim to answer two questions: 1) how to automatically collect the translation difficulty from BERTScore; and 2) how to integrate the difficulty into the score function. Figure 1 presents an overall illustration.

Pairwise Similarity Traditional n-gram overlap cannot extract semantic similarity, word embedding provides a means of quantifying the degree of overlap, which allows obtaining more accurate difficulty information. Since BERT is a strong language model, it can be utilized as a contextual embedding $\mathbf{O}_{\text{BERT}}$ (i.e., the output of BERT) for obtaining the representations of the reference $\mathbf{t}$ and the hypothesis $\mathbf{h}$. Given a specific hypothesis token h and reference token t, the similarity score $\text{sim}(t, h)$ is computed as follows:

$$\text{sim}(t, h) = \frac{\mathbf{O}_{\text{BERT}}(t)^{\top} \mathbf{O}_{\text{BERT}}(h)}{\|\mathbf{O}_{\text{BERT}}(t)\| \cdot \|\mathbf{O}_{\text{BERT}}(h)\|} \quad (1)$$

Subsequently, a similarity matrix is constructed by pairwise calculating the token similarity. Then the token-level matching score is obtained by greedily

Metric	En→De (All)			En→De (Top 30%)			De→En (All)			De→En (Top 30%)		
	$\|r\|$	$\|\tau\|$	$\|\rho\|$	$\|r\|$	$\|\tau\|$	$\|\rho\|$	$\|r\|$	$\|\tau\|$	$\|\rho\|$	$\|r\|$	$\|\tau\|$	$\|\rho\|$
BLEU	0.952	0.703	0.873	0.460	0.200	0.143	0.888	0.622	0.781	0.808	0.548	0.632
TER	0.982	0.711	0.873	0.598	0.333	0.486	0.797	0.504	0.675	**0.883**	0.548	0.632
METEOR	0.985	0.746	0.904	0.065	0.067	0.143	0.886	0.605	0.792	0.632	0.548	0.632
BERTScore	0.990	0.772	0.920	0.204	0.067	0.143	0.949	0.756	0.890	0.271	0.183	0.316
DA-BERTScore	**0.991**	**0.798**	**0.930**	**0.974**	**0.733**	**0.886**	**0.951**	**0.807**	**0.932**	0.693	**0.548**	**0.632**

Table 1: Absolute correlations with system-level human judgments on WMT19 metrics shared task. For each metric, higher values are better. Difficulty-aware BERTScore consistently outperforms vanilla BERTScore across different evaluation metrics and translation directions, especially when the evaluated systems are very competitive (i.e., evaluating on the top 30% systems).

searching for the maximal similarity in the matrix, which will be further taken into account in sentence-level score aggregation.

Difficulty Calculation The calculation of difficulty can be tailored for different metrics based on the overlap matching score. In this case, BERTScore evaluates the token-level overlap status by the pairwise semantic similarity, thus the token-level similarity is viewed as the bedrock of difficulty calculation. For instance, if one token (like "cat") in the reference may only find identical or synonymous substitutions in a few MT system outputs, then the corresponding translation difficulty weight ought to be larger than for other reference tokens, which further indicates that it is more valuable for evaluating the translation capability. Combined with BERTScore mechanism, it is implemented by averaging the token similarities across systems. Given K systems and their corresponding generated hypotheses $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K$, the difficulty of a specific token t in the reference $\mathbf{t}$ is formulated as

$$d(t) = 1 - \frac{\sum_{k=1}^{K} \max_{h \in \mathbf{h}_k} \mathrm{sim}(t, h)}{K} \quad (2)$$

An example is shown in Figure 1: the entity "cat" is improperly translated to "monkey" and "puppy", resulting in a lower pairwise similarity of the token "cat", which indicates higher translation difficulty. Therefore, by incorporating the translation difficulty into the evaluation process, the token "cat" is more contributive while the other words like "cute" are less important in the overall score.

Score Function Due to the fact that the translation generated by a current NMT model is fluent enough but not adequate yet, F-score which takes into account the *Precision* and *Recall*, is more appropriate to aggregate the matching scores, instead of only considering precision. We thus follow vanilla BERTScore in using F-score as the final score. The proposed method directly assigns difficulty weights to the counterpart of the similarity score **without any hyperparameter**:

$$\text{DA-}R_{\text{BERT}} = \frac{1}{|\mathbf{t}|} \sum_{t \in \mathbf{t}} d(t) \max_{h \in \mathbf{h}} \mathrm{sim}(t, h) \quad (3)$$

$$\text{DA-}P_{\text{BERT}} = \frac{1}{|\mathbf{h}|} \sum_{h \in \mathbf{h}} d(h) \max_{t \in \mathbf{t}} \mathrm{sim}(t, h) \quad (4)$$

$$\text{DA-}F_{\text{BERT}} = 2 \cdot \frac{\text{DA-}R_{\text{BERT}} \cdot \text{DA-}P_{\text{BERT}}}{\text{DA-}R_{\text{BERT}} + \text{DA-}P_{\text{BERT}}} \quad (5)$$

For any $h \notin \mathbf{t}$, we simply let $d(h) = 1$, i.e., retaining the original calculation. The motivation is that the human assessor keeps their initial matching judgement if the test taker produces a unique but reasonable alternative answer. We regard DA-F_{BERT} as the DA-BERTScore in the following part.

There are many variants of our proposed method: 1) designing more elaborate difficulty function (Liu et al., 2020; Zhan et al., 2021); 2) applying a smoothing function to the difficulty distribution; and 3) using other kinds of F-score, e.g., $F_{0.5}$-score. The aim of this paper is not to explore this whole space but simply to show that a straightforward implementation works well for MT evaluation.

4 Experiments

Data The WMT19 English↔German (En↔De) evaluation tasks are challenging due to the large discrepancy between human and automated assessments in terms of reporting the best system (Bojar et al., 2018; Barrault et al., 2019; Freitag et al., 2020). To sufficiently validate the effectiveness of

SYSTEM	BLEU ↑	TER ↓	METEOR ↑	BERTScore ↑	DA-BERTScore ↑	HUMAN ↑
Facebook.6862	0.4364 (⇓5)	0.4692 (⇓5)	0.6077 (⇓3)	0.7219 (⇓4)	**0.1555 (✓0)**	**0.347**
Microsoft.sd.6974	0.4477 (⇓1)	0.4583 (⇓1)	0.6056 (⇓3)	0.7263 (✓0)	0.1539 (⇓1)	0.311
Microsoft.dl.6808	0.4483 (⇑1)	0.4591 (⇓1)	0.6132 (⇑1)	0.7260 (✓0)	0.1544 (⇑1)	0.296
MSRA.6926	**0.4603 (⇑3)**	**0.4504 (⇑3)**	**0.6187 (⇑3)**	**0.7267 (⇑3)**	0.1525 (✓0)	0.214
UCAM.6731	0.4413 (✓0)	0.4636 (✓0)	0.6047 (⇓1)	0.7190 (⇓1)	0.1519 (⇓1)	0.213
NEU.6763	0.4460 (⇑2)	0.4563 (⇑4)	0.6083 (⇑3)	0.7229 (⇑2)	0.1521 (⇑1)	0.208
sum($\lvert\triangle_{\text{Rank}}\rvert$)	12	14	14	10	**4**	0

Table 2: Agreement of system ranking with human judgement on the top 30% systems (k=6) of WMT19 En→De Metrics task. ⇑/⇓ denotes that the rank given by the evaluation metric is higher/lower than human judgement, and ✓ denotes that the given rank is equal to human ranking. DA-BERTScore successfully ranks the best system that the other metrics failed. Besides, it also shows the lowest rank difference.

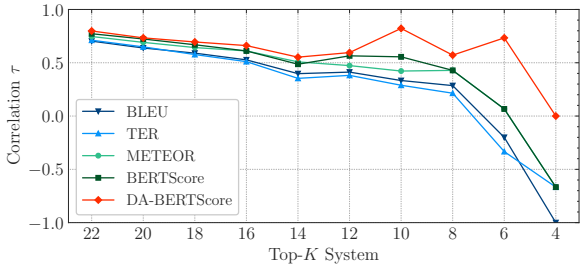


Figure 2: Effect of top-K systems in the En→De evaluation. DA-BERTScore is highly correlated with human judgment for different values of K, especially when all the systems are competitive (i.e., $K \leq 10$).

our approach, we choose these tasks as our evaluation subjects. There are 22 systems for En→De and 16 for De→En. Each system has its corresponding human assessment results. The experiments were centered on the correlation with system-level human ratings.

Comparing Metrics In order to compare with the metrics that have different underlying evaluation mechanism, four representative metrics: BLEU (Papineni et al., 2002), TER (Snover et al., 2006), METEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014), BERTScore (Zhang et al., 2020), which are correspondingly driven by n-gram, edit distance, word alignment and embedding similarity, are involved in the comparison experiments without losing popularity. For ensuring reproducibility, the original[1][2] and widely used implementation[3] was used in the experiments.

Main Results Following the correlation criterion adopted by the WMT official organization, Pearson's correlation r is used for validating the system-level correlation with human ratings. In addition, two rank-correlations Spearman's ρ and original Kendall's τ are also used to examine the agreement with human ranking, as has been done in recent research (Freitag et al., 2020). Table 1 lists the results. DA-BERTScore achieves competitive correlation results and further improves the correlation of BERTScore. In addition to the results on all systems, we also present the results on the top 30% systems where the calculated difficulty is more reliable and our approach should be more effective. The result confirms our intuition that DA-BERTScore can significantly improve the correlations under the competitive scenario, e.g., improving the $|r|$ score from 0.204 to 0.974 on En→De and 0.271 to 0.693 on De→En.

Effect of Top-K Systems Figure 2 compares the Kendall's correlation variation of the top-K systems. Echoing previous research, the vast majority of metrics fail to correlate with human ranking and even perform negative correlation when K is lower than 6, meaning that the current metrics are ineffective when facing competitive systems. With the help of difficulty weights, the degradation in the correlation is alleviated, e.g., improving τ score from 0.07 to 0.73 for BERTScore ($K = 6$). These results indicate the effectiveness of our approach, establishing the necessity for adding difficulty.

Case Study of Ranking Table 2 presents a case study on the En→De task. Existing metrics consistently select MSRA's system as the best system, which shows a large divergence from human judgement. DA-BERTScore ranks it the same as human (4th) because most of its translations have low difficulty, thus lower weights are applied in the scores. Encouragingly, DA-BERTScore ranks Facebook's system as the best one, which implies that it overco-

[1]https://www.cs.cmu.edu/ alavie/METEOR/index.html
[2]https://github.com/Tiiiger/bert_score
[3]https://github.com/mjpost/sacrebleu

	BERTS.	+DA	Sentence
Src	-	-	"I'm standing **right here** in front of you," one woman said.
Ref	-	-	„Ich stehe genau hier vor Ihnen ", sagte eine Frau.
MSRA	**0.9656**	0.0924	„Ich stehe **hier vor** Ihnen ", sagte eine Frau.
Facebook	0.9591	**0.1092**	„Ich stehe **hier direkt vor** Ihnen ", sagte eine Frau.
Src	-	-	France has more than 1,000 **troops on the ground** in the war-wracked country.
Ref	-	-	Frankreich hat über 1.000 Bodensoldaten in dem kriegszerstörten Land im Einsatz.
MSRA	**0.6885**	0.2123	Frankreich hat mehr als 1.000 **Soldaten vor Ort** in dem kriegsgeplagten Land.
Facebook	0.6772	**0.2414**	Frankreich hat mehr als 1000 **Soldaten am Boden** in dem kriegsgeplagten Land stationiert.

Table 3: Examples from the En→De evaluation. BERTS. denotes BERTScore. Words indicate the difficult translations given by our approach on the top 30% systems. DA-BERTScores are more in line with human judgements.

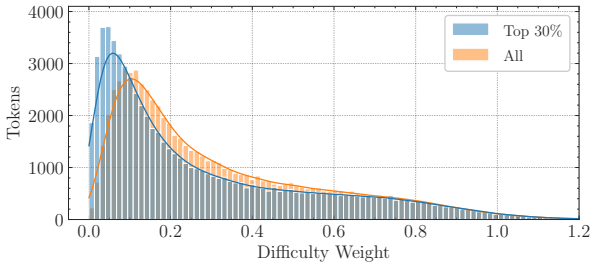


Figure 3: Distribution of token-level difficulty weights extracted from the En→De evaluation.

mes more challenging translation difficulties. This testifies to the importance and effectiveness of considering translation difficulty in MT evaluation.

Case Study of Token-Level Difficulty Table 3 presents two cases, illustrating that our proposed difficulty-aware method successfully identifies the omission errors ignored by BERTScore. In the first case, the Facebook's system correctly translates the token "right", and in the second case, uses the substitute "Soldaten am Boden" which is lexically similar to the ground-truth token "Bodensoldaten". Although the MSRA's system suffers word omissions in the two cases, its hypotheses receive the higher ranking given by BERTScore, which is inconsistent with human judgements. The reason might be that the semantic of the hypothesis is highly close to the reference, thus the slight lexical difference is hard to be found when calculating the similarity score. By distinguishing the difficulty of the reference tokens, DA-BERTScore successfully makes the evaluation focus on the difficult parts, and eventually correct the score of the Facebook's system, thus giving the right rankings.

Distribution of Difficulty Weights The difficulty weights can reflect the translation ability of a group of MT systems. If the systems in a group are of higher translation ability, the calculated difficulty weights will be smaller. Starting from this intuition, we visualize the distribution of difficulty weights as shown in Figure 3. Clearly, we can see that the difficulty weights are centrally distributed at lower values, indicating that most of the tokens can be correctly translated by all the MT systems. For the difficulty weights calculated on the top 30% systems, the whole distribution skews to zero since these competitive systems have better translation ability and thus most of the translations are easy for them. This confirms that the difficulty weight produced by our approach is reasonable.

5 Conclusion and Future Work

This paper introduces the conception of difficulty into machine translation evaluation, and verifies our assumption with a representative metric BERTScore. Experimental results on the WMT19 English↔German metric tasks show that our approach achieves a remarkable correlation with human assessment, especially for evaluating competitive systems, revealing the importance of incorporating difficulty into machine translation evaluation. Further analyses show that our proposed difficulty-aware BERTScore can strengthen the evaluation of word omission problems and generate reasonable distributions of difficulty weights.

Future works include: 1) optimizing the difficulty calculation; 2) applying to other MT metrics; and 3) testing on other generation tasks, e.g., speech recognition and text summarization.

Acknowledgement

This work was supported in part by the Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST). We thank the anonymous reviewers for their insightful comments.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Julian Chow, Lucia Specia, and Pranava Madhyastha. 2019. WMDO: Fluency-based word mover's distance for machine translation evaluation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 494–500, Florence, Italy. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Aaron L. F. Han, Derek F. Wong, and Lidia S. Chao. 2012. LEPOR: A robust evaluation metric for machine translation with augmented factors. In *Proceedings of COLING 2012: Posters*, pages 441–450, Mumbai, India. The COLING 2012 Organizing Committee.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. 2006. CDER: Efficient MT evaluation using block movements. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - A unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Dimitar Shterionov, Riccardo Superbo, Pat Nagle, Laura Casanellas, Tony O'Dowd, and Andy Way. 2018. Human versus automatic quality evaluation of NMT and PBSMT. *Machine Translation*, 32(3):217–235.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Runzhe Zhan, Xuebo Liu, Derek F. Wong, and Lidia S. Chao. 2021. Meta-curriculum learning for domain adaptation in neural machine translation. In *the 35th AAAI Conference on Artificial Intelligence, AAAI2021*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Uncertainty and Surprisal Jointly Deliver the Punchline: Exploiting Incongruity-Based Features for Humor Recognition

Yubo Xie, Junze Li, and Pearl Pu
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
{yubo.xie,junze.li,pearl.pu}@epfl.ch

Abstract

Humor recognition has been widely studied as a text classification problem using data-driven approaches. However, most existing work does not examine the actual joke mechanism to understand humor. We break down any joke into two distinct components: the set-up and the punchline, and further explore the special relationship between them. Inspired by the incongruity theory of humor, we model the set-up as the part developing semantic uncertainty, and the punchline disrupting audience expectations. With increasingly powerful language models, we were able to feed the set-up along with the punchline into the GPT-2 language model, and calculate the uncertainty and surprisal values of the jokes. By conducting experiments on the SemEval 2021 Task 7 dataset, we found that these two features have better capabilities of telling jokes from non-jokes, compared with existing baselines.

1 Introduction

One of the important aspects of computational humor is to develop computer programs capable of recognizing humor in text. Early work on humor recognition (Mihalcea and Strapparava, 2005) proposed heuristic-based humor-specific stylistic features, for example alliteration, antonymy, and adult slang. More recent work (Yang et al., 2015; Chen and Soo, 2018; Weller and Seppi, 2019) regarded the problem as a text classification task, and adopted statistical machine learning methods and neural networks to train models on humor datasets. However, only few of the deep learning methods have tried to establish a connection between humor recognition and humor theories. Thus, one research direction in humor recognition is to bridge the disciplines of linguistics and artificial intelligence.

In this paper, we restrict the subject of investigation to jokes, one of the most common humor types in text form. As shown in Figure 1, these jokes usually consist of a *set-up* and a *punchline*. The set-up creates a situation that introduces the hearer into the story framework, and the punchline concludes the joke in a succinct way, intended to make the hearer laugh. Perhaps the most suitable humor theory for explaining such humor phenomenon is the *incongruity theory*, which states that the cause of laughter is the perception of something incongruous (the punchline) that violates the hearer's expectation (the set-up).

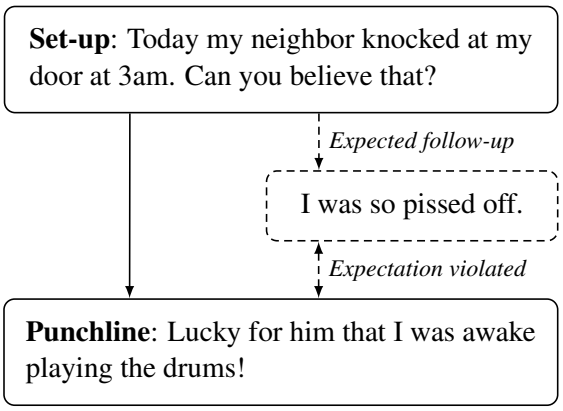


Figure 1: A joke example consisting of a set-up and a punchline. A violation can be observed between the punchline and the expectation.

Based on the incongruity theory, we propose two features for humor recognition, by calculating the degree of incongruity between the set-up and the punchline. Recently popular pre-trained language models enable us to study such relationship based on large-scale corpora. Specifically, we fed the set-up along with the punchline into the GPT-2 language model (Radford et al., 2019), and obtained the surprisal and uncertainty values of the joke, indicating how surprising it is for the model to generate the punchline, and the uncertainty while generating it. We conducted experiments on a manually labeled humor dataset, and the results showed that

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 33–39
August 1–6, 2021. ©2021 Association for Computational Linguistics

these two features could better distinguish jokes from non-jokes, compared with existing baselines. Our work made an attempt to bridge humor theories and humor recognition by applying large-scale pre-trained language models, and we hope it could inspire future research in computational humor.

2 Related Work

Humor Data Mihalcea and Strapparava (2005) created a one-liner dataset with humorous examples extracted from webpages with humor theme and non-humorous examples from Reuters titles, British National Corpus (BNC) sentences, and English Proverbs. Yang et al. (2015) scraped puns from the Pun of the Day website[1] and negative examples from various news websites. There is also work on the curation of non-English humor datasets (Zhang et al., 2019; Blinov et al., 2019). Hasan et al. (2019) developed UR-FUNNY, a multimodal humor dataset that involves text, audio and video information extracted from TED talks.

Humor Recognition Most of the existing work on humor recognition in text focuses on one-liners, one type of jokes that delivers the laughter in a single line. The methodologies typically fall into two categories: feature engineering and deep learning. Mihalcea and Strapparava (2005) designed three human-centric features (alliteration, antonymy and synonym) for recognizing humor in the curated one-liner dataset. Mihalcea et al. (2010) approached the problem by calculating the semantic relatedness between the set-up and the punchline (they evaluated 150 one-liners by manually splitting them into "set-up" and "punchline"). Shahaf et al. (2015) investigated funny captions for cartoons and proposed several features including perplexity to distinguish between funny and less funny captions. Morales and Zhai (2017) proposed a probabilistic model and leveraged background text sources (such as Wikipedia) to identify humorous Yelp reviews. Liu et al. (2018) proposed to model sentiment association between elementary discourse units and designed features based on discourse relations. Cattle and Ma (2018) explored the usage of word associations as a semantic relatedness feature in a binary humor classification task. With neural networks being popular in recent years, some deep learning structures have been developed for the recognition of humor in text. Chen and Lee (2017) and Chen and Soo (2018) adopted convolutional neural networks, while Weller and Seppi (2019) used a Transformer architecture to do the classification task. Fan et al. (2020) incorporated extra phonetic and semantic (ambiguity) information into the deep learning framework. In addition to these methodological papers, there are also some tasks dedicated to computational humor in recent years. SemEval 2020 Task 7 (Hossain et al., 2020) aims at assessing humor in edited news headlines. SemEval 2021 Task 7 (Meaney et al., 2021) involves predicting the humor rating of the given text, and if the rating is controversial or not. In this task, Xie et al. (2021) adopted the DeBERTa architecture (He et al., 2020) with disentangled attention mechanism to predict the humor labels.

Although the work of Mihalcea et al. (2010) is the closest to ours, we are the first to bridge the incongruity theory of humor and large-scale pre-trained language models. Other work (Bertero and Fung, 2016) has attempted to predict punchlines in conversations extracted from TV series, but their subject of investigation should be inherently different from ours—punchlines in conversations largely depend on the preceding utterances, while jokes are much more succinct and self-contained.

3 Humor Theories

The attempts to explain humor date back to the age of ancient Greece, where philosophers like Plato and Aristotle regarded the enjoyment of comedy as a form of scorn, and held critical opinions towards laughter. These philosophical comments on humor were summarized as the *superiority theory*, which states that laughter expresses a feeling of superiority over other people's misfortunes or shortcomings. Starting from the 18th century, two other humor theories began to challenge the dominance of the superiority theory: the *relief theory* and the *incongruity theory*. The relief theory argues that laughter serves to facilitate the relief of pressure for the nervous system (Morreall, 2020). This explains why laughter is caused when people recognize taboo subjects—one typical example is the wide usage of sexual terms in jokes. The incongruity theory, supported by Kant (1790), Schopenhauer (1883), and many later philosophers and psychologists, states that laughter comes from the perception of something incongruous that violates the expectations. This view of humor fits well the types of jokes commonly found in stand-up comedies,

[1] http://www.punoftheday.com/

where the set-up establishes an expectation, and then the punchline violates it. As an expansion of the incongruity theory, Raskin (1979) proposed the Semantic Script-based Theory of Humor (SSTH) by applying the semantic script theory. It posits that, in order to produce verbal humor, two requirements should be fulfilled: (1) The text is compatible with two different scripts; (2) The two scripts with which the text is compatible are opposite.

4 Methodology

The incongruity theory attributes humor to the violation of expectation. This means the punchline delivers the incongruity that turns over the expectation established by the set-up, making it possible to interpret the set-up in a completely different way. With neural networks blooming in recent years, pretrained language models make it possible to study such relationship between the set-up and the punchline based on large-scale corpora. Given the set-up, language models are capable of writing expected continuations, enabling us to measure the degree of incongruity, by comparing the actual punchline with what the language model is likely to generate.

In this paper, we leverage the GPT-2 language model (Radford et al., 2019), a Transformer-based architecture trained on the WebText dataset. We chose GPT-2 because: (1) GPT-2 is already pretrained on massive data and publicly available online, which spares us the training process; (2) it is domain independent, thus suitable for modeling various styles of English text. Our goal is to model the set-up and the punchline as a whole piece of text using GPT-2, and analyze the probability of generating the punchline given the set-up. In the following text, we denote the set-up as x, and the punchline as y. Basically, we are interested in two quantities regarding the probability distribution $p(y|x)$: uncertainty and surprisal, which are elaborated in the next two sections.

4.1 Uncertainty

The first question we are interested in is: given the set-up, how uncertain it is for the language model to continue? This question is related to SSTH, which states that, for a piece of text to be humorous, it should be compatible with two different scripts. To put it under the framework of set-up and punchline, this means the set-up could have multiple ways of interpretation, according to the following punchline. Thus, one would expect a higher uncertainty

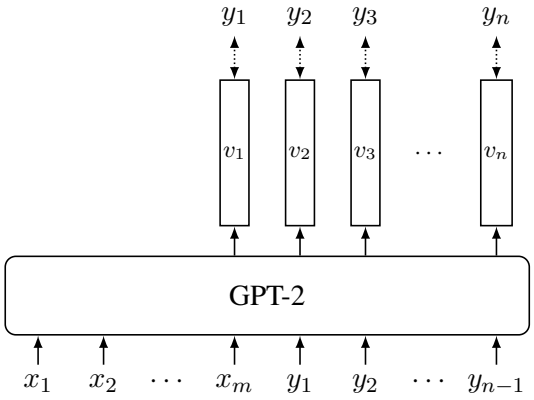


Figure 2: The set-up x and the punchline y are concatenated and fed into GPT-2 for predicting the next token. v_i's are probability distributions on the vocabulary.

value when the language model tries to continue the set-up and generate the punchline.

We propose to calculate the averaged entropy of the probability distributions at all token positions of the punchline, to represent the degree of uncertainty. As shown in Figure 2, the set-up x and the punchline y are concatenated and then fed into GPT-2 to predict the next token. While predicting the tokens of y, GPT-2 produces a probability distribution v_i over the vocabulary. The averaged entropy is then defined as

$$U(x,y) = -\frac{1}{|y|}\sum_{i=1}^{n}\sum_{w\in V} v_i^w \log v_i^w, \qquad (1)$$

where V is the vocabulary.

4.2 Surprisal

The second question we would like to address is: how surprising it is when the language model actually generates the punchline? As the incongruity theory states, laughter is caused when something incongruous is observed and it violates the previously established expectation. Therefore, we expect the probability of the language model generating the actual punchline to be relatively low, which indicates the surprisal value should be high. Formally, the surprisal is defined as

$$\begin{aligned} S(x,y) &= -\frac{1}{|y|}\log p(y|x) \\ &= -\frac{1}{|y|}\sum_{i=1}^{n}\log v_i^{y_i}. \end{aligned} \qquad (2)$$

5 Experiments

We evaluated and compared the proposed features with several baselines by conducting experiments

in two settings: predicting using individual features, and combining the features with a content-based text classifier.

5.1 Baselines

Similar to our approach of analyzing the relationship between the set-up and the punchline, Mihalcea et al. (2010) proposed to calculate the semantic relatedness between the set-up and the punchline. The intuition is that the punchline (which delivers the surprise) will have a minimum relatedness to the set-up. For our experiments, we chose two relatedness metrics that perform the best in their paper as our baselines, plus another similarity metric based on shortest paths in WordNet (Miller, 1995):

- **Leacock & Chodorow similarity** (Leacock and Chodorow, 1998), defined as

$$\text{Sim}_{lch} = -\log \frac{length}{2 * D}, \quad (3)$$

  where *length* is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of WordNet.

- **Wu & Palmer similarity** (Wu and Palmer, 1994) calculates similarity by considering the depths of the two synsets in WordNet, along with the depth of their *LCS* (Least Common Subsumer), which is defined as

$$\text{Sim}_{wup} = \frac{2 * \text{depth}(LCS)}{\text{depth}(C_1) + \text{depth}(C_2)}, \quad (4)$$

  where C_1 and C_2 denote synset 1 and synset 2 respectively.

- **Path similarity** (Rada et al., 1989) is also based on the length of the shortest path between two concepts in WordNet, which is defined as

$$\text{Sim}_{path} = \frac{1}{1 + length}. \quad (5)$$

In addition to the metrics mentioned above, we also consider the following two baselines related to the phonetic and semantic styles of the input text:

- **Alliteration**. The alliteration value is computed as the total number of alliteration chains and rhyme chains found in the input text (Mihalcea and Strapparava, 2005).

- **Ambiguity**. Semantic ambiguity is found to be a crucial part of humor (Miller and Gurevych, 2015). We follow the work of Liu et al. (2018) to compute the ambiguity value:

$$\log \prod_{w \in s} \text{num\_of\_senses}(w), \quad (6)$$

  where w is a word in the input text s.

5.2 Dataset

We took the dataset from SemEval 2021 Task 7.[2] The released training set contains 8,000 manually labeled examples in total, with 4,932 being positive, and 3,068 negative. To adapt the dataset for our purpose, we only considered positive examples with exactly two sentences, and negative examples with at least two sentences. For positive examples (jokes), the first sentence was treated as the set-up and the second the punchline. For negative examples (non-jokes), consecutive two sentences were treated as the set-up and the punchline, respectively.[3] After splitting, we cleaned the data with the following rules: (1) We restricted the length of set-ups and punchlines to be under 20 (by counting the number of tokens); (2) We only kept punchlines whose percentage of alphabetical letters is greater than or equal to 75%; (3) We discarded punchlines that do not begin with an alphabetical letter. As a result, we obtained 3,341 examples in total, consisting of 1,815 jokes and 1,526 non-jokes. To further balance the data, we randomly selected 1,526 jokes, and thus the final dataset contains 3,052 labeled examples in total. For the following experiments, we used 10-fold cross validation, and the averaged scores are reported.

5.3 Predicting Using Individual Features

To test the effectiveness of our features in distinguishing jokes from non-jokes, we built an SVM classifier (parameters can be found in Appendix A) for each individual feature (uncertainty and surprisal, plus the baselines). The resulted scores are reported in Table 1. Compared with the baselines, both of our features (uncertainty and surprisal) achieved higher scores for all the four metrics. In addition, we also tested the performance of uncertainty combined with surprisal (last row

[2] https://semeval.github.io/SemEval2021/

[3] We refer to them as set-up and punchline for the sake of convenience, but since they are not jokes, the two sentences are not real set-up and punchline.

	P	R	F1	Acc
Random	0.4973	0.4973	0.4958	0.4959
Sim_{lch}	0.5291	0.5179	0.4680	0.5177
Sim_{wup}	0.5289	0.5217	0.4919	0.5190
Sim_{path}	0.5435	0.5298	0.4903	0.5291
Alliteration	0.5353	0.5349	0.5343	0.5354
Ambiguity	0.5461	0.5365	0.5127	0.5337
Uncertainty	0.5840	0.5738	0.5593	0.5741
Surprisal	0.5617	0.5565	0.5455	0.5570
U+S	**0.5953**	**0.5834**	**0.5695**	**0.5832**

Table 1: Performance of individual features. Last row (U+S) is the combination of uncertainty and surprisal. P: Precision, R: Recall, F1: F1-score, Acc: Accuracy. P, R, and F1 are macro-averaged, and the scores are reported on 10-fold cross validation.

	P	R	F1	Acc
GloVe	0.8233	0.8232	0.8229	0.8234
GloVe+Sim_{lch}	0.8255	0.8251	0.8247	0.8250
GloVe+Sim_{wup}	0.8264	0.8260	0.8254	0.8257
GloVe+Sim_{path}	0.8252	0.8244	0.8239	0.8244
GloVe+Alliter.	0.8299	0.8292	0.8291	0.8297
GloVe+Amb.	0.8211	0.8203	0.8198	0.8201
GloVe+U	0.8355	0.8359	0.8353	0.8359
GloVe+S	0.8331	0.8326	0.8321	0.8326
GloVe+U+S	**0.8368**	**0.8368**	**0.8363**	**0.8365**

Table 2: Performance of the features when combined with a content-based classifier. U denotes uncertainty and S denotes surprisal. P: Precision, R: Recall, F1: F1-score, Acc: Accuracy. P, R, and F1 are macro-averaged, and the scores are reported on 10-fold cross validation.

of the table), and the resulting classifier shows a further increase in the performance. This suggests that, by jointly considering uncertainty and surprisal of the set-up and the punchline, we are better at recognizing jokes.

5.4 Boosting a Content-Based Classifier

Now that we have shown the advantage of our features when used individually in prediction, we would like to validate their effectiveness when combined with the commonly used word embeddings. Thus, we evaluated our features as well as the baselines under the framework of a content-based classifier. The idea is to see if the features could further boost the performance of existing text classifiers. To create a starting point, we encoded each set-up and punchline into vector representations by aggregating the GloVe (Pennington et al., 2014) embeddings of the tokens (sum up and then normalize by the length). We used the GloVe embeddings

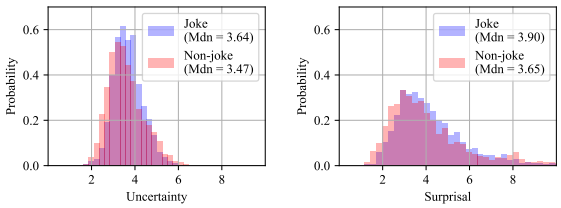


Figure 3: Histograms of uncertainty (left) and surprisal (right), plotted separately for jokes and non-jokes. Mdn stands for Median.

with dimension 50, and then concatenated the set-up vector and the punchline vector, to represent the whole piece of text as a vector of dimension 100. For each of the features (uncertainty and surprisal, plus the baselines), we appended it to the GloVe vector, and built an SVM classifier to do the prediction. Scores are reported in Table 2. As we can see, compared with the baselines, our features produce larger increases in the performance of the content-based classifier, and similar to what we have observed in Table 1, jointly considering uncertainty and surprisal gives further increase in the performance.

6 Visualizing Uncertainty and Surprisal

To get a straightforward vision of the uncertainty and surprisal values for jokes versus non-jokes, we plot their histograms in Figure 3 (for all 3,052 labeled examples). It can be observed that, for both uncertainty and surprisal, jokes tend to have higher values than non-jokes, which is consistent with our expectations in Section 4.

7 Conclusion

This paper makes an attempt in establishing a connection between the humor theories and the nowadays popular pre-trained language models. We proposed two features according to the incongruity theory of humor: uncertainty and surprisal. We conducted experiments on a humor dataset, and the results suggest that our approach has an advantage in humor recognition over the baselines. The proposed features can also provide insight for the task of two-line joke generation—when designing the text generation algorithm, one could exert extra constraints so that the set-up is chosen to be compatible with multiple possible interpretations, and the punchline should be surprising in a way that violates the most obvious interpretation. We hope our work could inspire future research in the community of computational humor.

References

Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of NAACL-HLT 2016*, pages 130–135.

Vladislav Blinov, Valeria Bolotova-Baranova, and Pavel Braslavski. 2019. Large dataset and language model fun-tuning for humor recognition. In *Proceedings of ACL 2019*, pages 4027–4032.

Andrew Cattle and Xiaojuan Ma. 2018. Recognizing humour using word associations and humour anchor extraction. In *Proceedings of COLING 2018*, pages 1849–1858.

Lei Chen and Chong Min Lee. 2017. Convolutional neural network for humor recognition. *CoRR*, abs/1702.02584.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of NAACL-HLT 2018, Volume 2 (Short Papers)*, pages 113–117.

Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Tongxuan Zhang. 2020. Phonetics and ambiguity comprehension gated attention network for humor recognition. *Complex.*, 2020:2509018:1–2509018:9.

Md. Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md. Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed (Ehsan) Hoque. 2019. UR-FUNNY: A multimodal language dataset for understanding humor. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2046–2056.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Nabil Hossain, John Krumm, Michael Gamon, and Henry A. Kautz. 2020. SemEval-2020 Task 7: Assessing humor in edited news headlines. In *Proceedings of SemEval@COLING 2020*, pages 746–758.

Immanuel Kant. 1790. Critique of judgment, ed. and trans. *WS Pluhar, Indianapolis: Hackett*.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and WordNet sense similarity for word sense identification. In *WordNet, An Electronic Lexical Database*. The MIT Press.

Lizhen Liu, Donghai Zhang, and Wei Song. 2018. Modeling sentiment association in discourse for humor recognition. In *Proceedings of ACL 2018, Volume 2 (Short Papers)*, pages 586–591.

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 Task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of SemEval@ACL 2021*.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of HLT/EMNLP 2005*, pages 531–538.

Rada Mihalcea, Carlo Strapparava, and Stephen G. Pulman. 2010. Computational models for incongruity detection in humour. In *Proceedings of CICLing 2010*, volume 6008 of *Lecture Notes in Computer Science*, pages 364–374.

George A. Miller. 1995. Wordnet: A lexical database for English. *Commun. ACM*, 38(11):39–41.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of English puns. In *Proceedings of ACL 2015*, pages 719–729.

Alex Morales and Chengxiang Zhai. 2017. Identifying humor in reviews using background text sources. In *Proceedings of EMNLP 2017*, pages 492–501.

John Morreall. 2020. Philosophy of Humor. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, fall 2020 edition. Metaphysics Research Lab, Stanford University.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and application of a metric on semantic nets. *IEEE Trans. Syst. Man Cybern.*, 19(1):17–30.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Raskin. 1979. Semantic mechanisms of humor. In *Annual Meeting of the Berkeley Linguistics Society*, volume 5, pages 325–335.

Arthur Schopenhauer. 1883. The world as will and idea (vols. i, ii, & iii). *Haldane, RB, & Kemp, J.(3 Vols.). London: Kegan Paul, Trench, Trubner*, 6.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. 2015. Inside jokes: Identifying humorous cartoon captions. In *Proceedings of SIGKDD 2015*, pages 1065–1074.

Orion Weller and Kevin D. Seppi. 2019. Humor detection: A transformer gets the last laugh. In *Proceedings of EMNLP-IJCNLP 2019*, pages 3619–3623.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of ACL 1994*, page 133–138.

Yubo Xie, Junze Li, and Pearl Pu. 2021. HumorHunter at SemEval-2021 Task 7: Humor and offense recognition with disentangled attention. In *Proceedings of SemEval@ACL 2021*.

	Running Time
Sim_{lch}	1.76 sec
Sim_{wup}	1.71 sec
Sim_{path}	1.71 sec
Alliteration	1.70 sec
Ambiguity	2.94 sec
Uncertainty	2.12 sec
Surprisal	2.49 sec
Uncertainty + Surprisal	2.26 sec

Table 3: Running time of the SVM classifiers trained on individual features.

	Running Time
GloVe	7.54 sec
GloVe + Sim_{lch}	14.85 sec
GloVe + Sim_{wup}	15.90 sec
GloVe + Sim_{path}	13.76 sec
GloVe + Alliteration	15.41 sec
GloVe + Ambiguity	14.28 sec
GloVe + Uncertainty	14.70 sec
GloVe + Surprisal	13.84 sec
GloVe + U + S	19.27 sec

Table 4: Running time of the content-based SVM classifiers combined with individual features. U denotes uncertainty and S denotes surprisal.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard H. Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of EMNLP 2015*, pages 2367–2376.

Dongyu Zhang, Heting Zhang, Xikai Liu, Hongfei Lin, and Feng Xia. 2019. Telling the whole story: A manually annotated chinese dataset for the analysis of humor in jokes. In *Proceedings of EMNLP-IJCNLP 2019*, pages 6401–6406.

A Model Parameters

For the SVM classifier, we set the regularization parameter $C = 1.0$, and used the RBF kernel with the kernel coefficient $\gamma = 1/n_{\text{features}}$. All models were trained and evaluated on a machine with Intel Core i7-6700K CPU, Nvidia GeForce GTX 1080 GPU, and 16GB RAM. The running time of each method is listed in Table 3 and Table 4.

Counterfactuals to Control Latent Disentangled Text Representations for Style Transfer

Sharmila Reddy Nangi[1] **Niyati Chhaya**[1] **Sopan Khosla**[2*]
Nikhil Kaushik[3*] **Harshit Nyati**[4*]
[1]Adobe Research, India [2]Carnegie Mellon University, USA
[3]Cohesity Storage Solutions, India [4]Adobe Systems, India
{snangi,nchhaya,hanyati}@adobe.com[1,4]
sopank@andrew.cmu.edu[2] nikhil.kaushik@cohesity.com[3]

Abstract

Disentanglement of latent representations into content and style spaces has been a commonly employed method for unsupervised text style transfer. These techniques aim to learn the disentangled representations and tweak them to modify the style of a sentence. In this paper, we propose a counterfactual-based method to modify the latent representation, by posing a 'what-if' scenario. This simple and disciplined approach also enables a fine-grained control on the transfer strength. We conduct experiments with the proposed methodology on multiple attribute transfer tasks like Sentiment, Formality and Excitement to support our hypothesis.

1 Introduction

Counterfactual Reasoning (Bottou et al., 2013) is leveraged in structured data analysis and econometrics towards generation of alternatives and estimation of alternate scenarios. Counterfactuals describe a causal situation of the form 'If X would have (not) occurred, Y would have (not) occurred' (Molnar, 2019). In interpretable machine learning, counterfactuals have been used to explain predictions of individual instances across various types of datasets and tasks (Neal et al., 2018; Martens and Provost, 2014; Wachter et al., 2017). Laugel et al.(2018) and Neal et al.(2018) use counterfactuals towards generating training data. Counterfactual reasoning also provides us with a unique ability to generate explanations and make causal analysis on the latent space. However, this technique has never been explored in natural language generation tasks. Here, we plug-in the concept of counterfactuals to the text-style transfer task, to enable the manipulation of latent spaces towards controlled transfer of style.

Existing works in text style transfer focus on transferring a specific target attribute. Unsupervised methods based on adversarial attacks (Fu et al., 2018; she), back translation (Prabhumoye et al., 2018), learning disentangled representations(John et al., 2019) have been popular in this domain. Other techniques include deletion of style-specific words and conditionally generate sentences in the target style (Li et al., 2018; Sudhakar et al., 2019). However, all of them fail to provide a control over the target style strength i.e. a clever manipulation of the latent space is non-trivial.

Recent works on controlled text generation include (Wang et al., 2019), which brings in a transformer-based model that modifies the gradient functions leading to controlled generation in the output space. Jin et al.(2019) is an unsupervised approach integrated during end-to-end model training. The drawback in all these efforts is the lack of a prefixed logic towards controlling the latent space. Our proposed method of counterfactuals fills in this gap and provides a logical method to control the latent spaces for enabling a smooth style transfer.

Our approach is based on the premise of disentangled representation spaces inspired from John et al.(2019). Separating out the style and content representations introduce an opportunity to fine-tune, resulting in the ability to control the output sentences specific to style. We **introduce a counterfactual reasoning module for controlling latent disentangled spaces for style transfer**. Figure 1 shows an illustrative example for the variants generated through our approach. To the best of our knowledge, this is the first work leveraging such a concept towards controlled text generation. Through extensive quantitative and qualitative experiments, across attributes and datasets, we conclude that the proposed approach is effective in providing control over the style strength and also shows that the best transfer performance is on par

*Work done while authors were at Adobe Research.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 40–48
August 1–6, 2021. ©2021 Association for Computational Linguistics

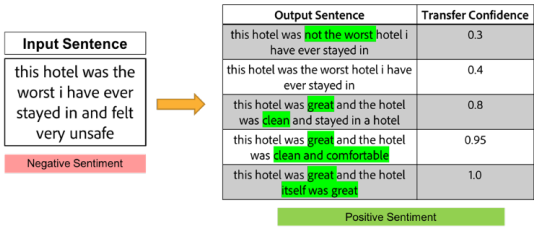


Figure 1: Example Counterfactuals showing the gradual 'control' introduced in the text style transfer.

with the existing baseline style-transfer techniques.

2 Approach

Figure 2 illustrates our proposed approach, that incorporates counterfactual reasoning to latent disentangled representations for manipulating style in text. It consists of (1) A Variational Autoencoder (VAE) model to learn the disentangled style and content representations for different stylistic attributes, (2) A Counterfactual Reasoning Module to control the latent representations for generating style variants.

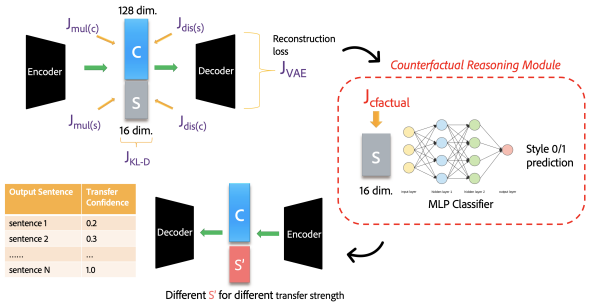


Figure 2: Proposed approach with Counterfactual Reasoning Module for Style Transfer

2.1 Learning Disentangled Representations

We adopt the model described in (John et al., 2019) for learning the disentangled content and style representations. Here, a VAE with an encoder-decoder is used to encode a sentence x into a latent distribution $H = q_E(h|x)$, guided by the loss function:

$$J_{VAE}(\theta_E, \theta_D) = J_{REC} + \lambda_{kl}\mathbb{KL}[q_E(h|x) \parallel p(h)]$$

where, θ_E and θ_D are the encoder and decoder parameters respectively. The first term encourages reconstruction, while the second term regularizes the latent space to a prior distribution $p(h)$ $(\mathcal{N}(0,1))$. We experiment with some variations of this architecture, which are detailed in section 3.

Additionally, Multi-Task $(J_{mul(s)}, J_{mul(c)})$ and Adversarial losses $(J_{adv(s)}, J_{adv(c)})$ are imposed on the latent space h to disentangle the embeddings into representing content c and style s, i.e., $h = [s; c]$, where $[;]$ denotes concatenation. These four losses ensure that the style and content information are present in, and *only* in their respective style(s) and content(c) embeddings.

Once we have the disentangled representations, our basic idea is to feed the generative model with the *same* content and a *different* style embedding to produce sentences of altering style. In (John et al., 2019), the average style embeddings of the target style is fed to the decoder. Intuitively, changing these style embeddings will produce different variants of target style sentences, but a disciplined approach to generate smooth style variants of the sentence is missing. We propose the counterfactual reasoning for this purpose.

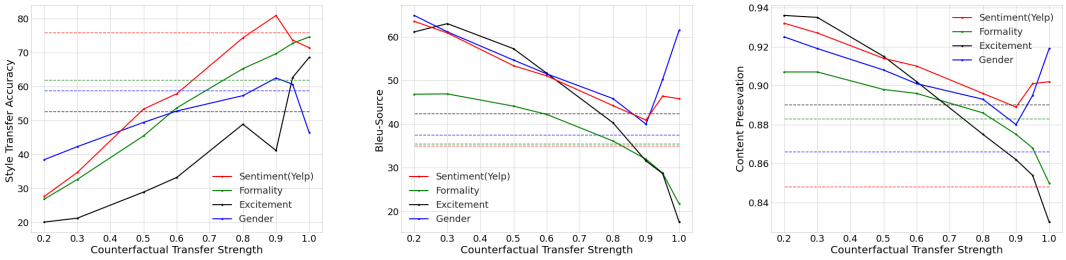
2.2 Counterfactual Reasoning Module

Counterfactuals (CF) are used for gradually changing the style representation along the target-style axis. A counterfactual explanation of an outcome Y takes the form 'if X had not occurred, Y would not have occurred'. We leverage this notion here. A Multi-layer Perceptron (MLP) classifier is trained on the disentangled style latent representations learnt by the VAE, such that every instance of style embedding s, predicts a target style (T) of a sentence. Now, the aim is to find s' such that it is close to s in the latent space but leads to a different prediction T', i.e. the target class. The CF generation loss is given by,

$$J_{cfactual} = L(s'|s) = \lambda(f_t(s') - p_t)^2 + L_1(s', s),$$

where t is the desired target style class for s', p_t is the probability with which we want to predict this target class (perfect transfer would mean $p_t = 1$), f_t is the model prediction on class t and L_1 is the distance between s' and s. The first term in the loss guides towards finding an s' that changes the model prediction to the target class and use of the L_1 distance ensures that minimum number of features are changed in order to change the prediction. λ is the weighting term. The resulting set of CFs are obtained by optimizing (Wachter et al., 2017) the following equation: $arg\min_{s'} \max_{\lambda} L(s'|s)$, subject to $|f_t(s' - p_t)| \leq \epsilon$ (tolerance parameter).

The CF generator is generalizable across different stylistic attributes. To generate multiple variants for a target style, CFs are generated varying



(a) Accuracy (b) Bleu-S (c) Cosine Similarity (CP)

Figure 3: Performance of the counterfactual model on multiple datasets. Style transfer accuracy (ACC) increases and the content preservation (BLEU-S, CP) decreases with increasing transfer strength.

the probability of target specific generation (or confidence), p_t. This results in different sentence variants with a similar target style but varied degrees for transfer strength. Finally, the disentangled representations enable finer control over the style dimensions with no risk of content loss during the counterfactual reasoning stage (as the content representations are retained).

3 Experiments

3.1 Proposed models

The VAE model adapted from (John et al., 2019), with RNN encoder-decoder blocks is R-VAE. We experiment with a variation by replacing RNNs with the transformer blocks (T-VAE). T-VAE-CF uses counterfactuals for generating variants, while models with -AVG use average style embedding of the target style to enable transfer. For T-VAE, we experimented with different loss combinations.-1,-2,-3,-4 refers to the inclusion of $J_{mul(s)}$, $J_{mul(s)} + J_{adv(s)}$, $J_{mul(s)} + J_{adv(s)} + J_{mul(c}$, $J_{mul(s)} + J_{adv(s)} + J_{mul(c)} + J_{adv(c)}$, respectively along with J_{VAE} in the overall loss function.

3.2 Baselines

We compare our best transfer models (with $p_t \approx 1$) against standard unsupervised style-transfer approaches. CrossAligned (CA)(Fu et al., 2018) aligns the hidden representations of original and style transferred sentences. T-D and T-DRG (Sudhakar et al., 2019) models delete attribute related words and conditionally generate words with the target style through transformer architecture.

3.3 Implementation

The counterfactual module has a linear classifier with a sigmoid activation, taking input dim. of 16 (s) and a output dim. 2 (style label). It is trained with Adam optimizer and 0.001 learning rate is used to minimize CCE loss. The transfer strength in CF-module, p_t, is varied from 0 to 1. Experiments with the following values $(0.2, 0.3, 0.5, 0.5, 0.8, 0.9, 0.95, 1.0)$ are reported.*

3.4 Datasets

We experiment with varied style attributes using 5 datasets. YELP is used for sentiment. Human gold standard references of these datasets from (Sudhakar et al., 2019) are used for evaluation. GYAFC dataset (Rao and Tetreault, 2018) is used for Formality and a new dataset GYAFC-excite with custom annotations for excitement is created†. POLITICAL (Voigt et al., 2018) and GENDER (Reddy and Knight, 2016)(similar to (Prabhumoye et al., 2018)) are used for the respective styles. The train-dev-test split as defined by original authors are used for all experiments.

3.5 Evaluation criteria

Style transfer accuracy (ACC) is measured by a dataset-specific Fasttext style classifier (Joulin et al., 2017). The classifiers report a % accuracy of 93.6, 87.6, 82.5, 78.3, 93.5 on the Yelp, GYAFC, GYAFC-Excite, Gender and Political datasets. *Content preservation* is measured through BLEU(Papineni et al., 2002) scores calculated against the source sentences(BLEU-S) and human references (BLEU-H), if available. We compute the cosine similarity (CP) to measure the vector-space similarity‡. *Language fluency* (PPL) is reported by

*Other implementation details, hyper-parameters, compute setup, and training times are provided in the appendix

†We cannot share the GYAFC-excitement dataset due to its license

‡Sentence embeddings for CP are calculated by concatenating the min, max, and mean of its word embeddings, ex-

Attribute⟶		Formality		Sentiment		Excitement	
Direction⟶		Formal⟶ Informal	Informal⟶ Formal	Positive ⟶ Negative	Negative ⟶ Positive	Less ⟶ More	More ⟶ Less
Source		it is **another way** to say that they don't like **you**	**hell yeah** for the first answer that girl answered for me	i always have a **great** dish here to eat	the wine was very **average** and the food was **even less**	it is a small **enjoyable** club	**wonderful** venue for tiff
Our Approach (CF Strength)	0.3	it is way to say it	yeah girl answer that question	i always have a *great* dish here to eat	the wine was very *average* and the food was *even good*	it 's a *good* club	*wonderful* venue
	0.5	you don't like it but it is way	yeah you should answer your question	i always have a *bad* dish here to eat .	the wine was very *average* and the food was *even better*	it 's a *great* club	*great* venue for tiff
	0.8	you can say it to you	oh girl answer that question	i always *have n't* been a though to go to order	the wine had *very unique* and the food was *excellent* too	it 's a *great* club	*good* venue for tiff
	0.9	you don't like it but it is way	oh my answer is *yes*	i always *do n't have a reviews* here to eat something .	the wine was very *reasonable* and the food was *even perfect*	it's a *great* club in vegas	*nice* venue
	0.95	u can say it to u	oh my answer is to answer that question	i always have a *bad* dish to eat here .	the wine had **very authentic** and the food was also **good**	it's a *great* club *in vegas*	*good* venue
	1.0	**u** can say **u r** a way	**oh my answer** is to answer that question	i **do n't always be having** a review to go here	the wine had *very unique* and the food was *excellent*	**absolutely loved** this club	**good** venue
Base	Avg	just say that way you don't know	answer the book for him , because i love that is what	i always do n't get home from a reviewer here	the wine was top notch and the food was even more	it is a small club and a fantastic museum	venue for wonderful for the after ballet

Table 1: Examples for Formality, Sentiment and Excitement with varying CF Strength using our framework.

MODEL	SENTIMENT(YELP)					FORMALITY				EXCITEMENT			
	Acc↑	Bleu-S↑	Bleu-H↑	CP↑	PPL↑	Acc↑	Bleu-S↑	CP↑	PPL↑	Acc↑	Bleu-S↑	CP↑	PPL↑
CA	76.6	47.95	37.15	0.92	-19.97	55.27	24.83	0.90	-19.08	78.25	33.43	0.87	-10.68
T-D	85.7	71.03	**54.08**	0.96	-20.12	46.55	**70.96**	0.95	-24.95	**83.85**	**69.04**	0.94	-13.52
T-DRG	77.4	70.60	54.00	0.96	-21.08	41.23	68.12	0.95	-26.91	74.15	63.65	**0.94**	-15.68
R-VAE-AVG	88.4	34.00	31.10	0.91	-15.08	69.02	32.78	0.90	**-15.18**	71.3	41.22	0.90	**-9.63**
R-VAE-CF	77.5	34.74	31.35	0.91	**-15.04**	62.17	32.47	0.91	-16.98	53.75	42.27	0.90	-9.83
T-VAE-AVG	76.9	34.39	29.19	0.88	-21.25	61.79	35.41	0.88	-23.05	52.55	42.36	0.89	-15.33
T-VAE-CF	**89.8**	34.61	29.49	0.88	-22.58	**74.64**	21.72	0.85	-23.74	68.6	17.57	0.83	-14.60

Table 2: Style Transfer Accuracy. Values for best performing models are reported in -CF variants.[For YELP p_t =: (T-VAE-4-CF,0.9); For FORMALITY(T-VAE-1-CF,1.0); For EXCITEMENT(T-VAE-1-CF,1.0)][‡‡]

MODEL	GENDER				POLITICAL			
	Acc	Bleu-S	CP	PPL	Acc	Bleu-S	CP	PPL
T-D	50.6	**82.50**	**0.97**	-39.05	74.0	**79.40**	**0.94**	-46.74
R-VAE-AVG	52.65	50.42	0.92	**-12.57**	100.0	10.56	0.86	**-26.65**
T-VAE-AVG	58.75	37.48	0.87	-18.22	**92.4**	33.25	0.88	-30.91
T-VAE-CF	**62.55**	39.99	0.88	-18.53	73.20	43.90	0.90	-30.17

Table 3: Gender & Political [For GENDER, p_t: (T-VAE-2-CF,0.9) .For POLITICAL:(T-VAE-2-CF,1.0)]

the perplexity of trigram KL-smoothed language model(Kneser and Ney, 1995), trained on the same corpus.

4 Results and Analysis

Transfer Control. Figure 3 shows the performance of CF variants across metrics for different styles. The CF generated variants from T-VAE-CF (solid lines) are compared against the reference values which take avg. embeddings (T-VAE-AVG) for target style (dotted lines). To recollect, the higher the CF transfer confidence (strength), the closer is the generated variant to the target attribute. Thus, the ideal performance is to have the highest accuracies for the highest CF confidence values (see figure 3(a)). Note that CF strength = 1 alludes to perfect transfer. This is difficult to achieve as CF in the representation space may not be generated for such a strict target. Hence, the variants generated with near perfect transfer target (CF strength = 0.8,0.9,0.95) show the best performance across metrics. The low transfer accuracies for models with low CF confidence establishes the ability of the model to stay near the source when the target strength is low. All models implemented with transfer control report improved performance w.r.t BLEU scores establishing the utility of the alternatives generator.

Table 2, 3 compares baselines with the proposed models. Note that the evaluation metrics for text style-transfer cannot be compared in isolation. There is always a trade-off between content preservation and transfer accuracy. Amongst the baselines, we observe that T-D and T-DRG report high content preservation with some loss in accuracy, but these models only cater to generating a single output sentence and there is no provision to generate the variants. Note that in most style dimensions, T-VAE based models show highest performance in transfer accuracy with good content preservation (CP), but, lower BLEU-S score. The lower BLEU-S scores indicates the ability of our model to generate variants that are not mere repetition of the input samples. R-VAE models show impressive perplexity values. For the

cluding stopwords(Fu et al., 2018)

political dataset, R-VAE baseline shows very high transfer accuracy but takes a tremendous hit in content preservation (BLEU), which is improved with the use of counterfactuals. Examples in Table 1 illustrate the gradual changes introduced by T-VAE-CF across different styles.

Human Evaluation: We conducted a crowdsourcing based experiment (through Amazon Mechanical Turk) to understand both - (A) How baselines compare to the generated text and (B) The interpretation of control as seen by human annotators. For the first experiment, the annotators were presented with sentences generated by our model, baselines and ground truth to evaluate and rank. Specifically, they were asked to score each of the output sentences on a Likert scale of range 1-5 across three aspects - transfer strength, content preservation and fluency. The key takeaways highlight that the sentences generated by our model are at par in terms of grammar and fluency and are better in terms of transfer control. As against text generated by baselines, the text generated by our proposed models is preferred by humans 70% of times (inter-annotator agreement 0.42).

For the second experiment to evaluate the control, we presented the sentence variants generated through different CFs (by varying p_t) and asked the annotators to rank them from best to worst based on their transfer strength. On an average, 60% individuals could grade the gradual control as intended by the model. If we bucket the sentences into low (with $p_t < 0.4$) and high groups (with $p_t > 0.7$), the annotators' preference for bucketing the output into the right confidence goes up to 73% on average (68% for low, and 81% for high), hence, confirming our hypothesis towards using CF for controlled generation.

5 Conclusion

We introduce the use of counterfactual reasoning towards controlling the latent disentangled representations for text style transfer. Experiments not only establish the superiority of the proposed models across standard metrics for a multitude of styles but also illustrate the utility of the gradual control variable in this model. We further validate the use for CF via a human evaluation establishing improved text attribute transfer.

References

Jennifer L Aaker. 1997. Dimensions of brand personality. *Journal of marketing research*, pages 347–356.

Martín Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(65):3207–3260.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. IMaT: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3097–3109, Hong Kong, China. Association for Computational Linguistics.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient

text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *ICASSP*, pages 181–184. IEEE Computer Society.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2018. Comparison-based inverse classification for interpretability in machine learning. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, pages 100–111, Cham. Springer International Publishing.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1453–1463. PMLR.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. Revision in continuous space: Unsupervised text style transfer without adversarial learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8376–8383.

David Martens and Foster Provost. 2014. Explaining data-driven document classifications. *MIS Q.*, 38(1):73–100.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119. Curran Associates, Inc.

Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. 2018. Open set learning with counterfactual images. In *The European Conference on Computer Vision (ECCV)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Sudha Rao. 2017. Are you asking the right questions? teaching machines to ask clarification questions. In *Proceedings of ACL 2017, Student Research Workshop*, pages 30–35, Vancouver, Canada. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 17–26, Austin, Texas. Association for Computational Linguistics.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. 2018. RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A VAE Models - Further Details

RNN-based (R-VAE). We adopt the model described in John et al.(2019) to disentangle the content and style representations with a recurrent neural network (RNN)-based VAE. The RNN encoder with Bi-GRUs (Cho et al., 2014) learns the hidden representation $q_E(h|x)$ by reading the input $x = (x_1, x_2, ..., x_n)$ sequentially. The RNN decoder, then decodes sequentially over time, predicting the probabilities of each token conditioned on the previous tokens and the latent representation. The reconstruction loss, which is the key loss for the generation objective, is the negative-log-likelihood loss as follows:

$$J_{REC} = \mathbb{E}_{h\sim q_E(h|x)}[-\sum_{t=1}^{n} \log P],$$

$$where \quad P = p(x_t|h, x_1, ..., x_{t-1})$$

The hidden space, h, is separated into 2 spaces while disentangling the style (s) and content (c) representations. Disentanglement is achieved using well-defined auxilliary losses.

Transformer-based (T-VAE). Transformers (Vaswani et al., 2017) have gained popularity for text generation due to their robust architectures. We introduce a transformer-based VAE inspired from Wang et al.(2019). The transformer encoder has a multi-headed self-attention block followed by a feed forward network (FFN). The decoder is similar to the encoder with an additional encoder-decoder attention block. Given an input sentence $x = (x_1, x_2, ..., x_n)$, the transformer encoder, E_{trans} learns a hidden word representation $(z_1, z_2, ..., z_n)$. They are pooled to get a sentence representation z, which is further encoded into a probabilistic latent space $q_E(h|x)$. A sample from this latent representation is given as an input to the encoder-decoder attention block in the decoder. The decoder reconstructs the input sentence x with condition on h. We adopt the label smoothing regularization (Li et al., 2020) while training, for performance improvement. The reconstruction loss (J_{REC}) is :

$$\mathbb{E}_{h\sim q_E(h|x)}[-\sum^{|x|}((1-\epsilon)\sum_{i=1}^{v}\bar{p_i}log(p_i)+ \frac{\epsilon}{v}\sum_{i=1}^{v}log(p_i))]$$

where, v is the vocabulary size, ϵ is the label smoothing parameter, p_i and $\bar{p_i}$ are the predicted and the ground truth probabilities over the vocabulary at every time step for word-wise decoding.

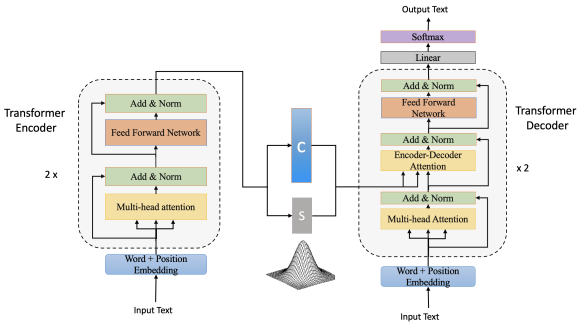


Figure 4: Transformer-based: T-VAE

KL Annealing. We also use an Adam optimiser and KL cost annealing technique (Bowman et al., 2016) to train our model. KL cost annealing refers to slow increase in the weight of the KL term (λ_{kl}) in the loss function from 0 to 1. This aids the training process as the model is warm-started to minimize the reconstruction loss in the initial iterations, followed by a gradual inclusion of KL loss term in the subsequent iterations.

A.1 Loss Functions

Auxiliary loss functions are used to achieve the text rewriting objectives. Note that the reconstruction loss is the primary loss generation but this does not take into consideration the style or the controlled generation.

We use Multi-task and Adversarial losses on the latent space h to disentangle the embeddings into representing content c and style s (i.e., $h = [s; c]$, where $[;]$ denotes concatenation) separately.

Style-oriented losses. Multitask Loss ensures that the style space s is discriminative for the style. We train a style classifier on s jointly with the autoencoder loss.

$$J_{mul(s)}(\theta_E; \theta_{mul(s)}) = -\sum_{l\in labels} t_s(l)\, log(y_s(l))$$

Dataset	Style	#train	#dev	#test	Source
Yelp	Positive	270K	2000	500	https://github.com/lijuncen/Sentiment-and-Style-Transfer/tree/master/data/yelp
	Negative	180K	2000	500	
GYAFC	Formal	48K	2000	950	https://github.com/raosudha89/GYAFC-corpus
	Informal	48K	2000	1250	
GYAFC-excitement	Exciting	36K	1990	1000	NA
	Non-Exciting	36K	1990	1000	
Political	Democrat	270K	2000	28K	http://tts.speech.cs.cmu.edu/style_models/political_data/
	Republican	270K	2000	28K	
Gender	Male	1.34M	2250	267K	http://tts.speech.cs.cmu.edu/style_models/gender_data/
	Female	1.34M	2250	267K	

Table 4: Datasets

where $\theta_{mul(s)}$ are the parameters for style multitask classifier, y_s is the style probability distribution predicted by the classifier and t_s is the ground truth style distribution.

Adversarial loss for style is introduced to ensure that the content space c is not-discriminative of the style. An adversarial classifier is trained, that deliberately discriminates the true style label using the content vector c, with the following loss.

$$J_{dis(s)}(\theta_{dis(s)}) = -\sum_{l \in labels} t_s(l) log(y_s^{'}(l))$$

where $\theta_{dis(s)}$ are the parameters for style adversary, $y_s^{'}$ is the style probability distribution predicted by the classifier on the content space.The encoder is then trained to learn a content vector space c, from which its adversary cannot predict style information. The objective is to maximize the cross entropy $H(p) = -\sum_{i \in labels} p_i log(p_i)$ with:

$$J_{adv(s)}(\theta_E) = H(y_s^{'}|c; \theta_{dis(s)})$$

Content-oriented losses. Multi-task loss aims to ensure that all content information is in the content space c. We define the content information using a bag-of-words (BoW) concept. Here, *part-of-speech* tags , i.e. *nouns* are used. (Liu et al., 2020; DBL) argue nouns in the text are considered as attribute-independent content. This definition allows a generic content loss for all style dimensions as against the previous work where content is defined as bag-of-words in a sentence, excluding stopwords and specific style (sentiment) related lexicon. The content multitask loss is analogical to style multitask loss as follows:

$$J_{mul(c)}(\theta_E; \theta_{mul(c)}) = -\sum_{w \in content} t_c(w)\, log(y_c(w))$$

Adversarial loss for content ensures that the style space does not contain content information. A classifier (content adversary), is trained on the style space to predict the content (BoW) features. Then similar to style, encoder is trained to learn s, from which this adversary cannot predict content information.

$$J_{dis(c)}(\theta_{dis(c)}) = -\sum_{w \in content} t_c(w) log({y_c}'(w))$$

$$J_{adv(c)}(\theta_E) = H({y_c}'|s; \theta_{dis(c)}),$$

Training with these losses along with reconstruction loss ensures that the latent space is disentangled, resulting in the final loss given by,

$$J_{total} = J_{VAE} + \lambda_{mul(s)} J_{mul(s)} - \lambda_{adv(s)} J_{adv(s)} + \lambda_{mul(c)} J_{mul(c)} - \lambda_{adv(c)} J_{adv(c)}$$

B Dataset details

The brief descriptions for datsets are as follows:
YELP: Reviews from Yelp. Each review is labeled with a sentiment class - positive or negative. The task is to change the label while rewriting.
GYAFC: Corpus created from a subset of Yahoo Answers. Each sample is tagged either formal or informal. The task is to switch the label.
GYAFC-Excitment: The task here is to convert the sentences from 'exciting' to 'non-exciting'. We create a subset of the GYAFC data where annotators (using Amazon Mechanical Turk), were asked to tag the sentence to be either showing excitement or not. Excitement follows the definition as given by (Aaker, 1997). We follow annotation scheme provided by Rao(2017).
POLITICAL: Comments from Facebook posts from United States Senate and House members. Each comment is labelled is with either Republican or Democrat tag. Task is to interchange between the two.
GENDER: Reviews from Yelp for food businesses. Each review is labeled with either male or female based on the author of the review. Task is to switch between the two.

Table 4 refers to the number of sentences in train-dev-test split available for each dataset. The URL

link to the data files are also provided for each of them.

C Implementation details

The dimensions of c and s are set to 128 and 16 respectively. The posterior probability distributions (μ, σ) learnt for the respective content and style also have the same dimensions. The learnt hidden state representation is converted to 128 (c) and 16 (s) with a linear layer.

For R-VAE, hidden state dimension is set to 256. For the T-VAE, the embedding size, latent layer and the self-attention layers all are set to 256. The inner dimension of FFN in the transformer is set to 1024. Each of the encoder and decoder is stacked with two layers of transformer blocks. We used the Adam optimizer for the VAE and the RMSProp optimizer for the discriminators, following stability tricks in adversarial training (Arjovsky and Bottou, 2017). Each optimizer has an initial learning rate of 10^{-3}. Models are trained for 50 epochs. Figure 4 illustrates the architecture of T-VAE.

Word embeddings initiated with word2vec (Mikolov et al., 2013) are trained on respective training sets. Both, the autoencoder and the discriminators are trained once per mini batch with $\lambda_{mul(s)}$, $\lambda_{mul(c)}$, $\lambda_{adv(s)}$, and $\lambda_{adv(c)} = 1$. The label smoothing parameter in the transformer loss ϵ is set to 0.1. The KL-Divergence penalty is weighted by $\lambda_{kl}(s)$ and $\lambda_{kl}(c)$ on style and content, respectively. During training, we also used the sigmoid KL annealing schedule

The hyper-parameter weights in the loss function $\lambda_{mul(s)}$, $\lambda_{mul(c)}$, $\lambda_{adv(s)}$, and $\lambda_{adv(c)}$ are chosen to be 1, as the values were Observed to be converging over iterations.

We implement our model based on Pytorch 0.4. We trained our models on a machine with 4 NVIDIA Tesla V100-SXM2-16GB GPUs. On a single GPU, our transformer model with all the losses (T-VAE-4) took approximately 0.4 s to train for one step with a batch of size 128. It takes around 10 hours to train our model on 1 GPU. Table 5 depicts the runtime details for all the model variations.

For our counterfactual generator model, we use the counterfactual model from Alibi library in Python[§]. On an average it takes 3 seconds to generate a counterfactual for a given input representation and transfer strength (p_t).

[§] Alibi Counterfactual Module

Dataset	Model	Batch Size	#batches in 1 epoch	Runtime for 1 epoch
Yelp	T-VAE-1	128	2375	247.32s
	T-VAE-2	128	2375	373.75s
	T-VAE-4	128	2375	1108.34s
Formality	T-VAE-1	32	3157	667.85s
	T-VAE-2	32	3157	944.97s
Excitement	T-VAE-1	64	1200	580.61s
	T-VAE-2	64	1200	602.99s
Gender	T-VAE-1	32	3156	333.58s
	T-VAE-2	32	3156	492.12s
Political	T-VAE-1	128	4233	751.92s
	T-VAE-2	128	4233	1050.30s

Table 5: Runtime details of model variations across different datasets

Dataset	Model	Counterfactual Module MLP Classifier	
		CCE Loss (Validation)	Accuracy (Validation)
Yelp	T-VAE-1	0.05	99.25
	T-VAE-2	0.04	99.31
	T-VAE-4	0.04	99.37
Formality	T-VAE-1	0.36	94.09
	T-VAE-2	0.33	97.43
Excitement	T-VAE-1	0.34	96.73
	T-VAE-2	0.22	96.87
Gender	T-VAE-1	0.11	96.17
	T-VAE-2	0.12	96.56
Political	T-VAE-1	0.005	99.992
	T-VAE-2	0.003	99.998

Table 6: Validation loss and accuracy for MLP classifier in counterfactual

Further details of our model summary and generated sentences are present here : https://bit.ly/34DYHP5

Attention Flows are Shapley Value Explanations

Kawin Ethayarajh
Stanford University
kawin@stanford.edu

Dan Jurafsky
Stanford University
jurafsky@stanford.edu

Abstract

Shapley Values, a solution to the credit assignment problem in cooperative game theory, are a popular type of explanation in machine learning, having been used to explain the importance of features, embeddings, and even neurons. In NLP, however, leave-one-out and attention-based explanations still predominate. Can we draw a connection between these different methods? We formally prove that — save for the degenerate case — attention weights and leave-one-out values cannot be Shapley Values. *Attention flow* is a post-processed variant of attention weights obtained by running the max-flow algorithm on the attention graph. Perhaps surprisingly, we prove that attention flows are indeed Shapley Values, at least at the layerwise level. Given the many desirable theoretical qualities of Shapley Values — which has driven their adoption among the ML community — we argue that NLP practitioners should, when possible, adopt attention flow explanations alongside more traditional ones.

1 Introduction

The approaches to model interpretability taken by the ML and NLP communities overlap in some areas and diverge in others. Notably, in machine learning, model prediction has sometimes been framed as a cooperative effort between the potential subjects of an explanation (e.g., input tokens) (Lundberg and Lee, 2017). But how should we allocate the credit for a prediction, given that some subjects contribute more than others (e.g., the sentiment words in sentiment classification)? The Shapley Value is a solution to this problem that uniquely satisfies several criteria for equitable allocation (Shapley, 1953). However, while Shapley Value explanations have been widely adopted by the ML community — to analyze the importance of features, neurons, and even training data (Ghorbani and Zou, 2019, 2020) — they have had far less traction in NLP, where leave-one-out and attention-based explanations still predominate.

What is the connection between these different paradigms? When, if ever, are attention weights and leave-one-out values effectively Shapley Values? The adoption of Shapley Values — which have their origins in game theory (Shapley, 1953) — by the ML community can be ascribed to their many desirable theoretical qualities. For example, consider a token whose masking out does not impact the model prediction in any way, regardless of how many other tokens in the sentence are also masked out. In game theory, such a token would be called a *null player*, whose Shapley Value is guaranteed to be zero (Myerson, 1977; Young, 1985). If we could provably identify the conditions under which attention weights and leave-one-out values are Shapley Values, we could extend such theoretical guarantees to them as well.

In this work, we first prove that — save for the degenerate case — attention weights and leave-one-out values cannot be Shapley Values. More formally, there is no set of *players* (i.e., possible subjects of an explanation, such as tokens) and *payoff* (i.e., function defining prediction quality) such that the values induced by attention or leave-one-out also satisfy the definition of a Shapley Value. We then turn to *attention flow*, a post-processed variant of attention weights obtained by running the max-flow algorithm on the attention graph (Abnar and Zuidema, 2020). We prove that when the players all come from the same layer (e.g., tokens in the input layer), there exists a payoff function such that attention flows are Shapley Values.

This means that under certain conditions, we can extend the theoretical guarantees associated with the Shapley Value to attention flow as well. As we show, these guarantees are axioms of faithful interpretation, and having them can increase

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 49–54
August 1–6, 2021. ©2021 Association for Computational Linguistics

confidence in interpretations of black-box NLP models. For this reason, we argue that whenever possible, NLP practitioners should use attention flow-based explanations alongside more traditional ones, such as gradients (Feng et al., 2018; Smilkov et al., 2017). We conclude by discussing some of the limitations in calculating Shapley Values for any arbitrary player set and payoff function in NLP.

2 Model Interpretation as a Game

The Shapley Value (Shapley, 1953) was proposed as a solution to a classic problem in game theory: When a group of players work together to achieve a payoff, how can we fairly allocate the payoff to each player, given that some contribute more than others? The players here are the potential subjects of the explanation (e.g., input tokens); the payoff is some quality of the model prediction (e.g., correctness). We contextualize the game theoretic terms with respect to model interpretability below.

Definition 2.1. A *player* is a possible subject of the explanation (e.g., character, token, embedding, neuron). $N = \{1, ..., n\}$ is the set of all players.

Definition 2.2. A *coalition* is a subset of players $S \subseteq N$ that work together. There are 2^n possible coalitions. The other players $N \setminus S$ are left out by being replaced with a non-subject that cannot affect the outcome (e.g., a zeroed-out embedding or a dropped-out neuron).

Definition 2.3. The *payoff* reflects some quality of the model prediction — e.g., correctness, confidence, entropy — made using a given coalition. It is defined by a *payoff function* $v : 2^N \rightarrow \mathbb{R}$, where $v(\emptyset) = 0$. The *value* $\phi_i(v)$ of a player i is the share of the payoff allocated to it. In other words, it is the importance accorded to subject i of an explanation.

Definition 2.4. A *game* is defined by (N, v), a player set N and payoff function v. It is a *transferable utility* game (TU-game), where the payoff can be distributed among the players as desired. In the game of model interpretation, the subjects of the explanation are framed as players working cooperatively to make the best possible prediction.

2.1 Equitable Allocation

How can we allocate the payoff equitably, in a way that reflects the actual contribution made by each player? In other words, how can we faithfully interpret a prediction? The game theory literature proposes that any equitable payoff allocation satisfies these three conditions (Myerson, 1977; Young, 1985; Ghorbani and Zou, 2019):

Condition 1. (*Null Player*): A player that induces no change in the payoff from joining any coalition has zero value. Formally, $\forall\ S \subseteq N \setminus \{i\}, v(S \cup \{i\}) = v(S) \implies \phi_i = 0$.

Condition 2. (*Symmetry*): Two players who induce the same change in payoff upon joining every coalition (that excludes them) have the same value. Formally, $\forall\ S \subseteq N \setminus \{i, j\}, v(S \cup \{i\}) = v(S \cup \{j\}) \implies \phi_i = \phi_j$.

Condition 3. (*Additivity*): The value of a player across two different games with payoff v, w should be the sum of its value in each game. Formally, $\forall\ i \in N, \phi_i(v + w) = \phi_i(v) + \phi_i(w)$.

2.2 The Shapley Value

The Shapley Value is a well-known solution to the problem of payoff allocation in a cooperative setting, as it uniquely satisfies the three criteria for equitable allocation in 2.1 (Shapley, 1953; Myerson, 1977; Young, 1985). It sets the value of a player to be its expected incremental contribution to a coalition, over all possible coalitions.

Definition 2.5. Where R is one of $n!$ possible permutations of the player set N, let $P_{R[:i]}$ be the subset of players that precede player i in the permutation. Then, for a given payoff function v, the Shapley Value of player i is

$$\phi_i(v) = \frac{1}{n!} \sum_R [v(P_{R[:i]} \cup \{i\}) - v(P_{R[:i]})] \quad (1)$$

There are other equivalent ways of expressing the Shapley Value, including as a sum over the 2^n possible coalitions.

In addition to satisfying our three criteria of equitable allocation (2.1), a Shapley Value distribution always exists and is unique for a TU-game (N, v). Unlike with attention weights, which have been criticized for allowing counterfactual explanations (Jain and Wallace, 2019; Serrano and Smith, 2019), there can thus be no counterfactual Shapley Value distribution for a given input and payoff function v. The distribution is also said to be *efficient*, since it allocates all of the payoff: $v(N) = \sum_{i \in N} \phi_i(v)$ (Myerson, 1977; Young, 1985). The Shapley Value can, in theory, be computed for any player set and payoff function. However, in practice, there are typically too many players to calculate this combinatorial expression exactly. Generally, estimates

are taken by uniformly sampling m random permutations $\mathcal{R}$ (Ghorbani and Zou, 2019):

$$\hat{\phi}_i(v) = \frac{1}{m} \sum_{R \in \mathcal{R}} [v(P_{R[:i]} \cup \{i\}) - v(P_{R[:i]})] \quad (2)$$

In the rest of this paper, we ask: Is there *some* TU-game (N, v) for which attention weights / attention flows / leave-one-out values are Shapley Values? If so, for which games?

3 Attention Weights

Many have argued that attention weights are not a faithful explanation, on the basis of *consistency* (i.e., poor correlation with other importance measures) and *non-exclusivity* (i.e., multiple explanations leading to the same outcome) (Jain and Wallace, 2019). Others have countered that they have some utility (Wiegreffe and Pinter, 2019). Without making assumptions about their inherent utility, we prove in this section that they cannot be Shapley Value explanations, outside of the degenerate case.

Proposition 1. If some player is attended to more than another, there is no TU-game (N, v) for which attention weights are Shapley Values.

Proof. Assume that attention weights are Shapley Values for some TU-game. Shapley Values are necessarily efficient (i.e., $v(N) = \sum_i \phi_i(v)$) (Myerson, 1977; Young, 1985), so for attention weights to be efficient, the only applicable payoff function would be the sum of attention weights. Since each player only has one Shapley Value for a given v, if it is attended to multiple times, its value must be the *total* attention paid to it: where $a_{j,i}$ denotes the attention j pays to i, $\phi_i(v) = \sum_{j \in N} a_{j,i}$. Note that the payoff for a coalition S is within some constant of its cardinality, since for a player j, the weights $a_{j,\cdot}$ of the players that it attends to sum to 1 (Bahdanau et al., 2015). We consider two cases.

Case 1 For a player j that attends to some other player, its contribution to the payoff of every $S \in N \setminus \{j\}$ is $\sum a_{j,\cdot} = 1$, implying $\phi_j(v) = 1$ by the Shapley Value definition (1). If some player (that pays attention) is more or less attended to than another — which is the point of using attention — this results in a contradiction. Thus ϕ_j cannot be the total attention paid to j.

Case 2 For a player i that doesn't attend to any other player, its contribution to the payoff of every $S \in N \setminus \{i\}$ is 0, since the attention paid to i is redistributed among other players when it is absent. This implies $\phi_i(v) = 0$ by (1). However, all input embeddings fall under this case, and we know at least one will be attended to; its attention weights will be non-zero, making this a contradiction. Thus ϕ_i cannot be the total attention paid to i. □

4 Attention Flows

What if we restricted the players to those from the same layer of a model? The remaining players still affect the prediction but can't have any of the payoff allocated to them. In this case, attention weights still cannot be Shapley Values. However, attention weights can be post-processed. Abnar and Zuidema (2020) proposed treating the self-attention graph as a flow network — where the attention weights are capacities — and then applying a max-flow algorithm (Ford and Fulkerson, 1956) to this network to calculate the maximum flow on each edge. We prove (by construction) that these *attention flows* are Shapley Values when the players are restricted to those from the same layer and the payoff is the total flow, as visualized in Figure 1.

Proposition 2. Consider a TU-game (N, v), where $N = \{1, ..., n\}$ players are all from the same layer. Let f denote the flow obtained by running a max-flow algorithm on the graph defined by the self-attention matrix, where the capacities are the attention weights. Let $v(S) = |f(S)|$, the *value of the flow* when only permitting flow through players in the coalition $S \subseteq N$. Then for each player i, its total outflow $|f_o(i)|$ is its Shapley Value.

Proof. Blocking the flow through a player $i \in S$ decreases $v(S)$ by that player's outflow $|f_o(i)|$, since the attention flow is only calculated once — with the entire graph — and not for each possible subgraph. Since the players are all disjoint and have no connections, blocking the flow through one player does not affect the outflow of any of the other players. This would not be the case, for example, if the players were in different layers, in which case changes in flow upstream would cause changes in flow downstream. Then for any coalition $S \subseteq N$ and player $i \notin S$, $v(S \cup \{i\}) = v(S) + |f_o(i)|$. We can rewrite the total outflow for player i as

$$\begin{aligned} |f_o(i)| &= v(S \cup \{i\}) - v(S), \forall\, S \subseteq N \\ &= \frac{n!}{n!} v(S \cup \{i\}) - v(S), \forall\, S \subseteq N \\ &= \frac{1}{n!} \sum_R \left[v(P_{R[:i]} \cup \{i\}) - v(P_{R[:i]}) \right] \end{aligned}$$

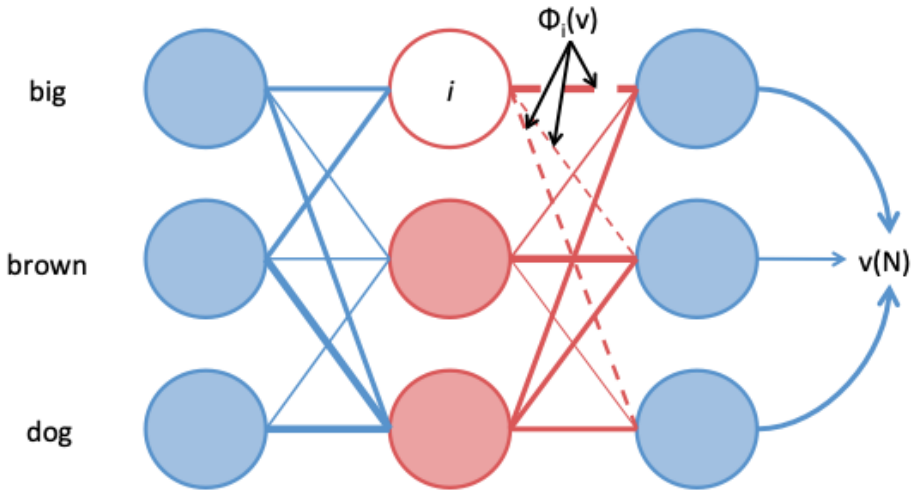


Figure 1: The attention flow network for three tokens across three layers, with player nodes (red) and non-player nodes (blue). The payoff $v(N)$ is the total flow through the network. $\phi_i(v)$ is the total outgoing flow of player i. Note that if we remove player i, then the total flow will decrease by $\phi_i(v)$, but the outgoing flow of the other two players (red) will stay the same. In other words, the contribution of player i to the total flow $v(N)$ is always $\phi_i(v)$; therefore, $\phi_i(v)$ is its Shapley Value. This construction is possible because the players are all in the same layer and therefore parallel; if one depended on another, then its outgoing flow could not be its Shapley Value.

which is just the Shapley Value definition (1). Note that the players cannot be from different layers — at least for the definition of v as the total flow value — because the Shapley Value distribution would not be efficient (i.e., $v(N) \neq \sum_{i \in N} \phi_i(v)$) and efficiency necessarily holds for Shapley Values. This in turn implies that the theoretical properties that hold for Shapley Values extend to attention flows under these conditions. □

Attention Rollout Abnar and Zuidema (2020) also proposed another post-processed variant of attention called *attention rollout*, in which the attention weight matrices from each layer are multiplied with those before it to get aggregated attention values. Attention roll-out values cannot be Shapley Values, however; this can be shown with a trivial extension of the proof to Proposition 1.

5 Leave-One-Out

Erasure describes a class of interpretability methods that aim to understand the importance of a representation, token, or neuron by erasing it and recording the resulting effect on model prediction (Li et al., 2016; Arras et al., 2017; Feng et al., 2018; Serrano and Smith, 2019). Although the Shapley Value technically falls under this class, most erasure-based methods only remove one entity — the one whose importance they want to estimate — and this only takes two forward passes, compared to $O(2^n)$ passes for the Shapley Value. Since only one entity is erased, this simpler group of erasure-based methods is called *leave-one-out* (Jain and Wallace, 2019; Abnar and Zuidema, 2020). We show in this section that leave-one-out values are not Shapley Values, except in the degenerate case.

Proposition 3. If $\exists\ i \in N$ such that player i is not a null player even when excluding the coalition $N \setminus \{i\}$, then there is no TU-game (N, v) for which leave-one-out values are Shapley Values.

Proof. Let the leave-one-out value of player i be denoted by $\text{LOO}_i(v)$. Let R' denote any permutation of N where $P_{R'[:i]} \neq N \setminus \{i\}$. By definition,

$$\begin{aligned}\phi_i(v) &= \frac{1}{n!} \sum_R \left[v(P_{R[:i]} \cup \{i\}) - v(P_{R[:i]}) \right] \\ &= \frac{1}{n!} \sum_{R'} \left[v(P_{R'[:i]} \cup \{i\}) - v(P_{R'[:i]}) \right] \\ &\quad + \frac{1}{n} \underbrace{\left(v(N) - v(N \setminus \{i\}) \right)}_{\text{LOO}_i(v)}\end{aligned}$$

By our assumption, the first term is non-zero, so there is no equivalence with $\text{LOO}_i(v)$. In practice, this assumption is almost always satisfied. □

Note that leave-one-out tells us very little about player importance for discrete payoff functions.

For example, if the payoff were the correctness (i.e., 1 if correct and 0 otherwise), then the importance of a player would be binary: it would either be critically important to prediction or totally irrelevant. This provides an incomplete picture — while there is enough redundancy in BERT-based models to tolerate some missing embeddings, this does not mean those embeddings are of no importance (Kovaleva et al., 2019; Ethayarajh, 2019; Michel et al., 2019). For example, if two representations played a critical and identical role in a prediction — but only one was necessary — then leave-one-out would assign each a value of zero, despite both being important. In contrast, the Shapley Value of both players would be non-zero and identical.

6 Applications

Because Shapley Values have many useful applications, attentions flows — and any other score that meets the criteria for a Shapley Value — have many useful applications as well:

- For one, using the various properties of the Shapley Value, we can provide more specific interpretations of model behavior than is currently the case, backed by theoretical guarantees. For example, if a token has zero attention flow in layer k but non-zero flow in layer $k-1$, then we can conclude that all the information it contains about the label (e.g., sentiment) was extracted by the model prior to the kth layer; this derives from the "null player" property of the Shapley Value. The same could not be said if the token only had a leave-one-out value of zero, since leave-one-out values are not Shapley Values.

- Interpretability in NLP often takes a single token or embedding to be the unit of analysis (i.e., a "player" in game theoretic terms). However, what if we wanted to understand the role of entire groups of tokens rather than individual ones? For most interpretability methods, there is no canonical way to aggregate scores across multiple units — we cannot necessarily add the raw attention scores of two tokens, since the usefulness of one may depend on the other. If we used a method that provided Shapley Values, we could easily redefine a "player" to be a group of tokens, such that all tokens in the same player group would simultaneously be included or excluded from a coalition.

- Recent work has used the Data Shapley — an extension of the Shapley Value — to estimate the contribution of each example in the training data to a model's decision boundary (Ghorbani and Zou, 2019). If we're fine-tuning BERT for sentiment classification, for example, we might want to know which sentence is more helpful: "This movie was great!" or "This was better than I expected." We can answer such questions by using the Data Shapley. To our knowledge, this has been done in computer vision but not in NLP.

7 Limitations and Future Work

Because Shapley Values — and by extension, attention flows — have many theoretical guarantees that are axioms of faithful interpretation, we encourage NLP practitioners to provide attention flow-based explanations alongside more traditional ones. This is not without limitations, however. As proven in Proposition 2, this equivalence only holds for a specific payoff function — the total flow through a layer — which is reflective of model confidence but not of the prediction correctness.

But why do we need attention flows at all if, in theory, Shapley Values can be calculated for any arbitrary player set and payoff function? While this is true in theory, because of the combinatorial calculation (1), it is computationally intractable in most cases. While it is possible to take a Monte Carlo estimate (2), in practice the bounds can be quite loose (Maleki et al., 2013). Finding TU-games for which the Shapley Value can be calculated exactly in polynomial time — as with attention flow -– is an important line of future work. These explanations may come with trade-offs: for example, SHAP is a kind of Shapley Value that assumes contributions are linear (i.e., a coalition can't be greater than the sum of its parts), which makes it much faster to calculate but restricts the set of possible payoff functions (Lundberg and Lee, 2017). Still, such methods will be critical to providing explanations that are both fast and faithful.

Acknowledgements

We thank Rishi Bommasani and the reviewers for their helpful feedback. KE was supported by an NSERC PGS-D and the Stanford Institute for Human-Centered AI.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Lester Randolph Ford and Delbert R Fulkerson. 1956. Maximal flow through a network. *Canadian journal of Mathematics*, 8:399–404.

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251.

Amirata Ghorbani and James Zou. 2020. Neuron shapley: Discovering the responsible neurons. *arXiv preprint arXiv:2002.09815*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

Roger B Myerson. 1977. Graphs and cooperation in games. *Mathematics of operations research*, 2(3):225–229.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

Lloyd Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, pages 31–40.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20.

H Peyton Young. 1985. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14(2):65–72.

Video Paragraph Captioning as a Text Summarization Task

Hui Liu, Xiaojun Wan
Wangxuan Institute of Computer Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{xinkeliuhui, wanxiaojun}@pku.edu.cn

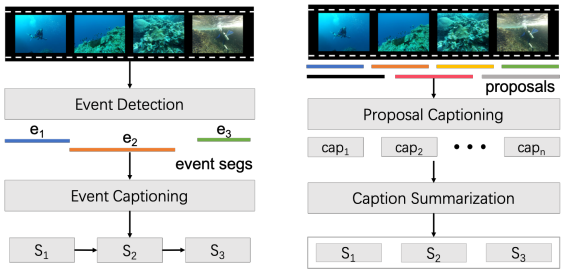
Abstract

Video paragraph captioning aims to generate a set of coherent sentences to describe a video that contains several events. Most previous methods simplify this task by using ground-truth event segments. In this work, we propose a novel framework by taking this task as a text summarization task. We first generate lots of sentence-level captions focusing on different video clips and then summarize these captions to obtain the final paragraph caption. Our method does not depend on ground-truth event segments. Experiments on two popular datasets ActivityNet Captions and YouCookII demonstrate the advantages of our new framework. On the ActivityNet dataset, our method even outperforms some previous methods using ground-truth event segment labels.

1 Introduction

Video captioning, the task of describing the content of a video in natural language, is a popular task both in computer vision and natural language processing. In the beginning, researchers try to generate sentence-level captions for short video clips (Venugopalan et al., 2015). Krishna et al. (2017) propose the task of dense video captioning. The system needs to detect event segments first and then generate captions. Park et al. (2019) propose the task of video paragraph captioning: they use ground-truth event segments and focus on generating coherent paragraphs. Lei et al. (2020) follow the task setting and propose a recurrent transformer model that can generate more coherent and less repetitive paragraphs. Considering the ground-truth event segments are often unavailable in practice, our goal is to generate paragraph captions without ground-truth segments.

The conventional framework of video paragraph captioning is shown in Figure 1a. Given an untrimmed video, an Event Detection module outputs a set of non-redundant event segments. The Event Captioning module generates captions for these segments. The works of (Park et al., 2019; Zhou et al., 2019; Lei et al., 2020) use ground-truth event segments and focus on the Event Captioning module. Zhou et al. (2019) use extra human-annotated bounding boxes as supervision. (Sah et al., 2017; Zhou et al., 2018; Mun et al., 2019) use predicted event segments and generate captions based on them. Sah et al. (2017) also summarizes these captions to generate a paragraph. The above methods heavily depend on accurate event segments. According to previous works (Zhou et al., 2018; Mun et al., 2019), the performance of the Event Detection module is not so good, making it a performance bottleneck. To tackle this problem, we propose a novel framework VPCSum as shown in Figure 1b. For a given video, we first extract dense event segment candidates (we call proposals), and a Proposal Captioning module is used to generate proposal captions. Then we treat video paragraph captioning as a text summarization task to obtain the final summary (paragraph caption).



(a) Conventional Framework (b) Our Framework

Figure 1: Comparison between conventional framework and ours.

In this work, we only consider extractive summarization, where the paragraph caption is composed by selecting from proposal captions. We conduct experiments on two popular datasets ActivityNet

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 55–60
August 1–6, 2021. ©2021 Association for Computational Linguistics

Captions and YouCookII. The results demonstrate the advantages of our framework. On the ActivityNet Captions dataset, our method even outperforms some previous methods using ground-truth event segment labels.

2 Our VPCSum Method

As illustrated in Figure 1b, our framework has three modules. **Proposal Extraction**: it extracts dense proposals for a video; **Proposal Captioning**: it generates captions for extracted proposals; **Caption Summarization**: it summarizes the generated proposal captions to obtain the video paragraph caption. We will introduce each module next.

2.1 Proposal Extraction

For proposal extraction, we use the BMN model (Lin et al., 2019), a popular model for temporal action proposal generation. It can extract complete and accurate proposals. We extract the top 100 proposals for each video.

2.2 Proposal Captioning

For proposal captioning, we choose the TSRM-RNN model (Wang et al., 2020) for ActivityNet Captions and VTransformer model (Lei et al., 2020) for YouCookII according to proposal captioning performance. We believe that if we choose a better sentence-level captioning model, the performance can be further improved.

2.3 Caption Summarization

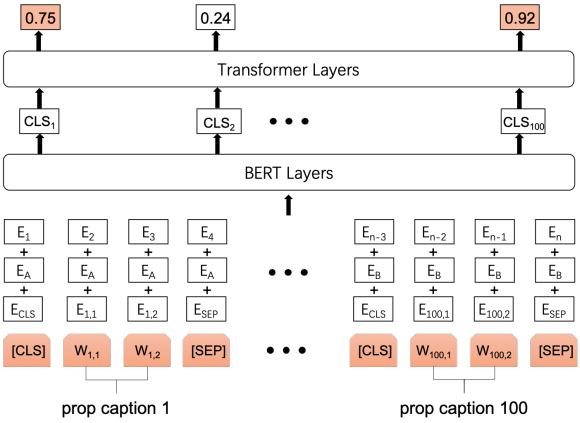


Figure 2: Architecture of the caption summarization model.

The caption summarization module summarizes proposal captions to generate the final video paragraph caption. In this work, we focus on extractive summarization. The architecture of our summarization model is illustrated in Figure 2. We first sort the proposal captions according to the proposal start time and add special [CLS] and [SEP] tokens to the beginning and end of each caption. We use the summation of token embeddings, segment embeddings, and position embeddings to represent each word. The input representations are fed into a pre-trained BERT model (Devlin et al., 2018), after which we obtain the contextual token representations. We use the contextual vectors of [CLS]s to represent each caption and feed them into stacked transformer layers (Vaswani et al., 2017). We use a sigmoid layer to compute the score of each caption:

$$x_i = \sigma(W h_i^L + b) \quad (1)$$

where W and b are trainable parameters, h_i^L is the vector for caption i from the top transformer layer.

For extractive summarization, we need to annotate each sentence according to the gold summary as our training target. Many researchers use a greedy algorithm (Nallapati et al., 2016), sentences are selected one by one to maximize the ROUGE score against the gold summary. The selected sentences are labeled 1 while others are labeled 0 (hard-label). In our task, we find a more effective soft-label annotation method. We label caption c_i with the max ROUGE score against gold captions and use binary cross-entropy as our loss function:

$$y_i = \max_{g_j \in gold} ROUGE(c_i, g_j) \quad (2)$$

$$\mathcal{L} = -\sum_i (y_i \log x_i + (1 - y_i) \log(1 - x_i)) \quad (3)$$

where g_j is the j-th gold caption.

2.4 Leverage Visual Information

The above caption summarization module assigns each proposal caption a predicted score, indicating how likely it appears in the final paragraph caption. The predicted score only depends on text information. To leverage visual information, we need a "visual summarization" module, which gives a visually weighting score to each proposal. The ESGN model (Mun et al., 2019) seems a good choice for us. It uses a pointer network to select events from proposals and assigns a visually weighting score for each proposal. We use this model to compute the visually weighting score.

Now we can extract the final paragraph caption. The final score of each proposal caption is a

weighted sum of the textually weighting score s_{txt} and the visually weighting score s_{vis}:

$$score(i) = s_{txt,i} + \lambda s_{vis,i} \quad (4)$$

where λ is a hyper-parameter tuned on validation set. We select captions according to $score(i)$ and use Trigram Blocking to reduce redundancy, as in Liu and Lapata (2019).

3 Experiments

3.1 Datasets

We conduct experiments on ActivityNet Captions (Krishna et al., 2017) and YouCookII (Zhou et al., 2017). ActivityNet Captions contains 10,009 videos in train set, 4,917 videos in val set. Each video has 3.65 event segments on average. Following (Lei et al., 2020), the original val set is split into ae-val with 2,460 videos for validation and ae-test with 2,457 videos for test. YouCookII contains 1,333 videos in train set, 457 videos in val set. Each video has 7.70 event segments on average.

3.2 Evaluation Metrics

Following (Lei et al., 2020; Park et al., 2019), we evaluate the captioning performance at paragraph level. We report standard caption metrics, including BLEU@4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), CIDEr (Vedantam et al., 2015). We also evaluate repetition using R@4 (Xiong et al., 2018). We use the scripts provided by (Lei et al., 2020) for evaluation[1].

3.3 Implementation Details

For video preprocessing, we use appearance and optical flow features provided by Zhou et al. (2018). For BMN model and captioning models, we use the same hyperparameters suggested by the authors. For ESGN model, we use a transformer encoder instead of an RNN encoder, with hidden size set to 512, number of heads set to 8, number of layers set to 3. For our caption summarization model, we use the base BERT model, 2 stacked transformer layers with hidden size set to 768, number of heads set to 8. We set max input length to 1,700, batch size to 10, λ to 1 for ActivityNet Captions and max input length to 1,000, batch size to 1, λ to 1 for YouCookII. Warmup steps are set to step num of 1 epoch. We use Adam optimizer with an initial learning rate of $6e-4$.

[1] https://github.com/jayleicn/recurrent-transformer

3.4 Baselines and Results

We compare our **VPCSum** model with the following baselines. **Soft-NMS**: it uses Soft-NMS (Bodla et al., 2017) to select event segments from BMN proposals, and uses the proposal captioning model to generate captions; **ESGN**: similar to Soft-NMS, but it uses ESGN model (Mun et al., 2019) to select event segments from BMN proposals; **V-Trans**: a Vanilla Transformer model, proposed by (Zhou et al., 2018); **Trans-XL**: a Transformer-XL model, proposed by (Lei et al., 2020); **MART**: a recurrent transformer model (Lei et al., 2020); **COOT**: it uses pretrained features to train MART model (Ging et al., 2020). Originally, the last four models deal with ground-truth event segments. For fair comparison, we also test them with predicted event segments generated by ESGN model [2].

Models	B@4	M	C	R@4↓
Soft-NMS	10.33	14.93	22.58	10.17
ESGN	10.38	15.74	21.85	6.51
V-Trans	9.89	15.11	20.95	7.04
Trans-XL	10.36	14.89	20.73	7.45
MART	10.13	14.94	20.16	6.09
COOT	9.85	14.67	21.83	7.15
VPCSum	**10.89**	**15.84**	**24.33**	**1.54**
V-trans*	9.31	15.54	21.33	7.45
Trans-XL*	10.25	14.91	21.71	8.79
MART*	9.78	15.57	22.16	**5.44**
COOT*	**10.85**	**15.99**	**28.19**	6.64

Table 1: Comparison with baselines on ActivityNet Captions ae-test split. * means the model uses ground-truth event segments. We report BLEU@4 (B@4), METEOR (M), CIDEr (C), Repetition (R@4).

Tables 1 and 2 show the results on ActivityNet Captions and YouCookII. We can observe that on the ActivityNet Captions, our model **VPCSum** within the new framework can generate better paragraph captions with higher Bleu@4, METEOR, and CIDEr and lower repetition score R@4, even outperforming V-trans*, Trans-XL*, MART* models using ground-truth event segments on every metric. On the YouCookII dataset, our model outperforms the models in the same setting but is inferior to the models using ground-truth segments. This may be because YouCookII has more segments

[2] We use the codes and pretrained models provided by the authors and only replace ground-truth event segments with ESGN predicted event segments.

Models	B@4	M	C	R@4↓
Soft-NMS	5.58	13.67	18.18	4.94
ESGN	5.36	13.37	17.01	2.82
V-Trans	5.35	13.37	16.88	2.85
Trans-XL	4.78	12.67	14.24	3.20
MART	5.61	13.44	16.56	4.63
COOT	5.96	14.21	19.67	5.99
VPCSum	**6.14**	**15.11**	**23.92**	**0.65**
V-trans*	7.62	15.65	32.26	7.83
Trans-XL*	6.56	14.76	26.35	6.30
MART*	8.00	15.90	35.74	**4.39**
COOT*	**9.44**	**18.17**	**46.06**	6.30

Table 2: Comparison with baselines on YouCookII val split.

(7.70 vs 3.65) than ActivityNet Captions.

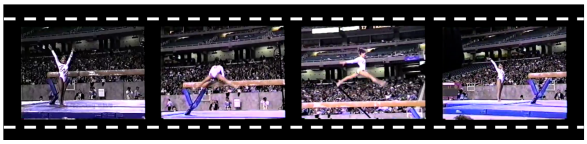
3.5 Ablation Study

Table 3 shows the ablation study on ActivityNet Captions. Compared to our full model (Full), the traditional extractive summarization annotation method (Hard-label) is not suitable for our task. If we set λ in Eq.(4) to 0 (w/o vis), the model loses useful visual information and performs not well. If we remove Trigram Blocking (w/o tri-blk), the performance also degrades and repetition becomes a problem (R@4 increases to 7.91). To verify the role of pretrained BERT model, we retrain our VPCSum without BERT pretrained weights (w/o pretrain). We can see that BERT pretrained weights are not the major factor to the final performance. We also replace our summarization model with unsupervised methods LexRank (Erkan and Radev, 2004) and LSA(Steinberger and Jezek, 2004). The results show that simple unsupervised summarization methods cannot handle our data well and supervised training is necessary.

Models	B@4	M	C	R@4↓
Full	10.89	15.84	24.33	1.54
Hard-label	10.29	14.99	21.71	1.19
w/o vis	10.68	15.78	23.34	1.36
w/o tri-blk	10.46	15.61	21.40	7.91
w/o pretrain	10.84	15.81	24.00	1.55
LexRank	7.78	13.65	14.19	26.51
LSA	7.24	14.48	12.43	28.14

Table 3: Model ablation study on ActivityNet Captions ae-test split.

3.6 Qualitative Results



Ground Truth: A girl jumps onto a balance beam. She does a gymnastics routine on the balance beam. She does a flip off the balance beam and lands on a mat.
MART: A gymnast is seen standing ready with her arms up and leads into her performing a gymnastics routine. She continues performing several flips and tricks and tricks and ends with her jumping down and walking away. She continues her routine and ends with her jumping down and jumping down and walking away.
MART*: A gymnast is seen standing before a beam and begins performing a gymnastics routine. The girl then performs a routine on the beam and ends with her jumping down and jumping down and jumping. The girl jumps off the beam and lands on the mat and jumps off the beam.
VPCSum: A gymnast is seen standing ready with her arms up and begins to do a routine. She does a gymnastics routine on the beam. She dismounts and lands on the mat.

Figure 3: An example from ActivityNet Captions.

We show an example in Figure 3 with paragraph captions generated by MART, MART* and our VPCSum model. Compared to other models, our model can generate more clear and correct sentences with less redundancy. The generated paragraph of our model can better describe the process of the whole event.

3.7 Human Evaluation

	Ours	MART	Ours	MART*
rel.	56.0%†	44.0%†	52.7%	47.3%
div.	56.7%†	43.3%†	56.7%†	43.3%†

Table 4: Human evaluation results. Statistically significant differences ($p < 0.05$) are marked with †.

We also conduct a human evaluation on randomly sampled 50 videos from the ActivityNet Captions val set. The annotators are asked to choose the better caption from two models in two aspects: **relevance** (how related is the caption to the video content) and **diversity** (how diverse is the generated text). We compare our VPCSum model with MART and MART* respectively. We have 17 college students as our annotators. Each video is judged by 3 annotators. We show the results of the pairwise experiments in Table 4. Our VPCSum model performs better in relevance and diversity,

and more people choose the caption of our model as the better one.

4 Conclusion

In this work, we view the task of video paragraph captioning as a text summarization task and propose a novel framework VPCSum. It allows us to use text summarization techniques to handle this challenging task. Experimental results on two popular datasets show the advantages of our model. In the future, we will explore using abstractive summarization methods to generate better video paragraph captions.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), MSRA Collaboration Research Project (FY20-Research-Sponsorship-266) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. 2017. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. Coot: Cooperative hierarchical transformer for video-text representation learning. *arXiv preprint arXiv:2011.00597*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. 2020. Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. *arXiv preprint arXiv:2005.05402*.

Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. 2019. Streamlined dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6588–6597.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *arXiv preprint arXiv:1611.04230*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. 2019. Adversarial inference for multi-sentence video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6598–6608.

Shagan Sah, Sourabh Kulhare, Allison Gray, Subhashini Venugopalan, Emily Prud'Hommeaux, and Raymond Ptucha. 2017. Semantic text summarization of long videos. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 989–997. IEEE.

Josef Steinberger and Karel Jezek. 2004. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4:93–100.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.

Teng Wang, Huicheng Zheng, and Mingjing Yu. 2020. Dense-captioning events in videos: Sysu submission to activitynet challenge 2020. *arXiv preprint arXiv:2006.11693*.

Yilei Xiong, Bo Dai, and Dahua Lin. 2018. Move forward and tell: A progressive generator of video descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 468–483.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. 2019. Grounded video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6578–6587.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2017. Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv:1703.09788*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.

Are VQA Systems RAD?
Measuring Robustness to Augmented Data with Focused Interventions

Daniel Rosenberg and **Itai Gat** and **Amir Feder** and **Roi Reichart**
Technion - Israel Institute of Technology
daniel.rnberg@gmail.com | {itaigat@ | feder@campus. | roiri@}technion.ac.il

Abstract

Deep learning algorithms have shown promising results in visual question answering (VQA) tasks, but a more careful look reveals that they often do not understand the rich signal they are being fed with. To understand and better measure the generalization capabilities of VQA systems, we look at their robustness to counterfactually augmented data. Our proposed augmentations are designed to make a focused intervention on a specific property of the question such that the answer changes. Using these augmentations, we propose a new robustness measure, Robustness to Augmented Data (RAD), which measures the consistency of model predictions between original and augmented examples. Through extensive experimentation, we show that RAD, unlike classical accuracy measures, can quantify when state-of-the-art systems are not robust to counterfactuals. We find substantial failure cases which reveal that current VQA systems are still brittle. Finally, we connect between robustness and generalization, demonstrating the predictive power of RAD for performance on unseen augmentations.[1]

1 Introduction

In the task of Visual Question Answering (VQA), given an image and a natural language question about the image, a system is required to answer the question accurately (Antol et al., 2015). While the accuracy of these systems appears to be constantly improving (Fukui et al., 2016; Yang et al., 2016; Lu et al., 2016), they are sensitive to small perturbations in their input and seem overfitted to their training data (Kafle et al., 2019).

To address the problem of overfitting, the VQA-CP dataset was proposed (Agrawal et al., 2018). It is a reshuffling of the original VQA dataset, such that the distribution of answers per question type (e.g., "what color", "how many") differs between the train and test sets. Using VQA-CP, Kafle et al. (2019) demonstrated the poor out-of-distribution generalization of many VQA systems. While many models were subsequently designed to deal with the VQA-CP dataset (Cadene et al., 2019; Clark et al., 2019; Chen et al., 2020; Gat et al., 2020), aiming to solve the out-of-distribution generalization problem in VQA, they were later demonstrated to overfit the unique properties of this dataset (Teney et al., 2020). Moreover, no measures for robustness to distribution shifts have been proposed.

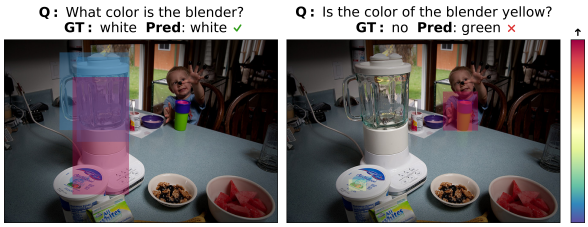


Figure 1: Predictions and attention maps of a state-of-the-art VQA-CP model over a VQA example (left) and its augmentation (right). A robust model should use the information it utilizes in the original example to correctly answer the augmented one.

In this work we propose a consistency-based measure that can indicate on the robustness of VQA models to distribution shifts. Our robustness measure is based on counterfactual data augmentations (CADs), which were shown useful for both training (Kaushik et al., 2019) and evaluation (Garg et al., 2019; Agarwal et al., 2020). CADs are aimed at manipulating a specific property while preserving all other information, allowing us to evaluate the robustness of the model to changes to this property.

For example, consider transforming a "what color" question to a "yes/no" question, as depicted in Figure 1. The counterfactual reasoning for such a transformation is: "what would be the question if it had a yes/no answer?". While VQA models

[1] Our code and data are available at: https://danrosenberg.github.io/rad-measure/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 61–70
August 1–6, 2021. ©2021 Association for Computational Linguistics

have seen many of both question types, their combination (yes/no questions about color) has been scarcely seen. If a model errs on such a combination, this suggests that to answer the original question correctly, the model uses a spurious signal such as the correlation between the appearance of the word "color" in the question and a particular color in the answer (e.g. here, color ⇒ white). Further, this example shows that some models cannot even identify that they are being asked a "yes/no" question, distracted by the word "color" in the augmented question and answering "green".

Our robustness measure is named RAD: Robustness to (counterfactually) Augmented Data (Section 2.1). RAD receives (image, question, answer) triplets, each augmented with a triplet where the question and answer were manipulated. It measures the consistency of model predictions when changing a triplet to its augmentation, i.e., the robustness of the model to (counterfactual) augmentations. We show that using RAD with focused interventions may uncover substantial weaknesses to specific phenomenon (Section 3.2), namely, users are encouraged to precisely define their interventions such that they create *counterfactual* augmentations. As a result, pairing RAD values with accuracy gives a better description of model behavior.

In general, to effectively choose a model in complex tasks, complementary measures are required (D'Amour et al., 2020). Thus, it is important to have interpretable measures that are widely applicable. Note that in this work we manipulate only textual inputs - questions and answers, but RAD can be applied to any dataset for which augmentations are available. In particular, exploring visual augmentations would be beneficial for the analysis of VQA systems. Further, representation-level counterfactual augmentations are also valid, which is useful when generating meaningful counterfactual text is difficult (Feder et al., 2020).

Our augmentations (CADs) are generated semi-automatically (Section 2.2), allowing us to directly intervene on a property of choice through simple templates. As in the above example, our augmentations are based on compositions of two frequent properties in the data (e.g., "what color" and "yes/no" questions), while their combination is scarce. Intuitively, we would expect a model with good generalization capacities to properly handle such augmentations. While this approach can promise coverage of only a subset of the examples in the VQA and VQA-CP datasets, it allows us to control the sources of the model's prediction errors.

We conduct extensive experiments and report three key findings. First, for three datasets, VQA, VQA-CP, and VisDial (Das et al., 2017), models with seemingly similar accuracy are very different in terms of robustness, when considering RAD with our CADs (Section 3). Second, we show that RAD with alternative augmentation methods, which prioritize coverage over focused intervention, cannot reveal the robustness differences. Finally, we show that measuring robustness using RAD with our CADs predicts the accuracy of VQA models on unseen augmentations, establishing the connection between robustness to our controlled augmentations and generalization (Section 4).

2 Robustness to Counterfactuals

In this section, we first present RAD (Section 2.1), which measures model consistency on question-answer pairs and their augmented modifications. Then, we describe our template-based CAD generation approach (Section 2.2), designed to provide control over the augmentation process.

2.1 Model Robustness

We denote a VQA dataset with $\mathcal{U} = \{(x_v, x_q, y) \in \mathcal{V} \times \mathcal{Q} \times \mathcal{Y}\}$, where x_v is an image, x_q is a question and y is an answer. We consider a subset $\mathcal{D} \subseteq \mathcal{U}$ for which we can generate augmentations. For an example $(x_v, x_q, y) \in \mathcal{D}$, we denote an augmented example as $(x_v, x'_q, y') \in \mathcal{D}'$. In this paper we generate a single augmentation for each example in $\mathcal{D}$, resulting in a one-to-one correspondence between $\mathcal{D}$ and the dataset of modified examples $\mathcal{D}'$. We further define $J(\mathcal{D}; f)$ as the set of example indices for which a model f correctly predicts y given x_v and x_q.

RAD assesses the proportion of correctly answered modified questions, among correctly answered original questions, and is defined as,

$$\mathrm{RAD}(\mathcal{D}, \mathcal{D}'; f) = \frac{|J(\mathcal{D}; f) \cap J(\mathcal{D}'; f)|}{|J(\mathcal{D}; f)|}. \quad (1)$$

Note that RAD is in $[0, 1]$ and the higher the RAD of f is, the more robust f is.

As original examples and their augmentations may differ in terms of their difficulty to the model, it is important to maintain symmetry between $\mathcal{D}$ and $\mathcal{D}'$. We hence also consider the backward view

of RAD, defined as $\mathrm{RAD}(\mathcal{D}', \mathcal{D}; f)$. For example, "yes/no" VQA questions are easier to answer compared to "what color" questions, as the former have two possible answers while the latter have as many as eight. Indeed, state-of-the-art VQA models are much more accurate on yes/no questions compared to other question types (Yu et al., 2019). Hence, if "what color" questions are augmented with "yes/no" counterfactuals, we would not expect $\mathrm{RAD}(\mathcal{D}', \mathcal{D}; f) = 1$ as generalizing from "yes/no" questions ($\mathcal{D}'$) to "what color" questions ($\mathcal{D}$) requires additional reasoning capabilities.

RAD is not dependant on the accuracy of the model on the test set. A model may perform poorly overall but be very consistent on questions that it has answered correctly. Conversely, a model that demonstrates seemingly high performance may be achieving this by exploiting many dataset biases and be very inconsistent on similar questions.

2.2 Counterfactual Augmentations

In the VQA dataset there are three answer types: "yes/no", "number" (e.g., '2', '0') and "other" (e.g., 'red', 'tennis'), and 65 question types (e.g., "what color", "how many", "what sport"). In our augmentations, we generate "yes/no" questions from "number" and "other" questions.

For example, consider the question-answer pair "What color is the vehicle? Red", this question-answer pair can be easily transformed into "Is the color of the vehicle red? Yes". In general, "what color" questions can be represented by the template: "What color is the <*Subj*>? <*Color*>". To generate a new question, we first identify the subject (<*Subj*>) for every "what color" question, and then integrate it into the template "Is the color of the <*Subj*> <*Color*>? Yes". As the model was exposed to both "what color" and "yes/no" questions, we expect it to correctly answer the augmented question given that it correctly answers the original. Yet, this augmentation requires some generalization capacity because the VQA dataset contains very few yes/no questions about color.

Our templates are presented in Table 1 (see Table 6 in the appendix for some realizations). The augmentations are counterfactual since we intervene on the question type, a prior that many VQA systems exploit (Goyal et al., 2017), keeping everything else equal. The generation process is semi-automatic, as we had to first manually specify templates that would yield augmented questions that we can expect the model to answer correctly given that it succeeds on the original question.

	Original	Augmented
Y/N ← C	What color is the <*S*>? <*C1*>	Is the color of the <*S*> <*C2*>? Yes/No
Y/N ← HM	How many <*S*>? <*N1*>	Are there <*N2*> <*S*>? Yes/No
Y/N ← WK	What kind of <*S*> is this? <*O1*>	Is this <*S*> <*O2*>? Yes/No

Table 1: Our proposed template-based augmentations.

To achieve this goal, we apply two criteria: **(a)** The template should generate a grammatical English question; and **(b)** The generated question type should be included in the dataset, but not in questions that address the same semantic property as the original question. Indeed, yes/no questions are frequent in the VQA datasets, but few of them address color (first template), number of objects (second template), and object types (third template). When both criteria are fulfilled, it is reasonable to expect the model to generalize from its training set to the new question type.

Criterion (a) led us to focus on yes/no questions since other transformations required manual verification for output grammaticality. While we could have employed augmentation templates from additional question types into yes/no questions, we believe that our three templates are sufficient for evaluating model robustness. Overall, our templates cover 11% of the VQA examples (Section 3.1).

3 Robustness with RAD and CADs

In the following, we perform experiments to test the robustness of VQA models to augmentations. We describe the experimental setup, and evaluate VQAv2, VQA-CPv2, VisDial models, each on our augmentations and on other alternatives.[2]

3.1 Experimental Setup

Baseline Augmentations We compare our augmentations to three alternatives: VQA-Rephrasings (Reph, Shah et al., 2019), ConVQA (Ray et al., 2019), and back-translation (BT, Sennrich et al., 2016). VQA-Rephrasings is a manual generation method, where annotators augment each validation question with three re-phrasings. ConVQA is divided into the L-ConVQA and CS-ConVQA subsets. In both subsets, original validation examples are augmented to create new question-answer pairs. L-ConVQA is automatically generated based

[2]The URLs of the software and datasets, and the implementation details are all provided in Appendices C and D.

Dataset	Model\$\mathcal{D}'$	RAD$(\mathcal{D}, \mathcal{D}')$ (%)							Acc.
		Y/N ← C	Y/N ← HM	Y/N ← WK	BT	Reph	L-ConVQA	CS-ConVQA	
VQA-CP	RUBi	64.92	57.15	62.59	85.57	77.73	78.02	65.93	46.66
	LMH	1.01	22.82	50.10	83.68	75.04	64.54	50.65	53.72
	CSS	0.94	11.73	39.95	77.54	68.89	10.67	38.64	58.47
VQA	BUTD	67.15	58.68	78.59	87.43	79.28	75.78	70.19	63.09
	BAN	74.40	62.45	82.51	88.17	81.14	79.37	70.18	65.92
	Pythia	65.00	60.61	81.60	88.42	82.86	77.02	69.45	64.56
	VisualBERT	79.99	68.29	85.98	88.52	84.09	82.09	71.75	65.62
VisDial	FGA	31.36	57.69	-	91.42	-	-	-	53.07
	VisDialBERT	62.08	56.06	-	94.04	-	-	-	55.78

Table 2: RAD over our proposed augmentations (Y/N ← C, Y/N ← HM, Y/N ← WK) and alternatives (BT, Reph, ConVQA). The rows correspond to state-of-the-art models on VQA-CP (top), VQA (middle) and Visual Dialog (bottom). Reph and ConVQA were not created for VisDial, and it does not have "what kind" questions. The last column corresponds to validation accuracy.

Dataset	Model\$\mathcal{D}'$	Accuracy$(\mathcal{D})$ (%)						
		Y/N ← C	Y/N ← HM	Y/N ← WK	BT	Reph	L-ConVQA	CS-ConVQA
VQA-CP	RUBi	65.85	17.35	44.14	45.80	46.51	72.14	66.67
	LMH	68.87	44.24	50.58	52.35	53.78	65.07	61.76
	CSS	72.87	63.16	51.83	56.37	58.81	49.84	56.12
VQA	BUTD	79.44	54.43	63.49	60.37	62.23	75.05	62.42
	BAN	80.72	62.37	66.48	63.02	64.81	74.94	65.01
	Pythia	81.62	57.49	64.42	61.69	63.88	74.55	63.79
	VisualBERT	80.85	58.89	64.46	62.71	64.96	76.50	66.01
VisDial	FGA	55.62	40.00	-	61.53	-	-	-
	VisDialBERT	68.99	50.77	-	63.47	-	-	-

Table 3: Original accuracy over our proposed augmentations (Y/N ← C, Y/N ← HM, Y/N ← WK) and alternatives (BT, Reph, ConVQA). The rows correspond to state-of-the-art models on VQA-CP (top), VQA (middle) and Visual Dialog (bottom). Reph and ConVQA were not created for VisDial, and it does not have "what kind" questions.

on scene graphs attached to each image, and CS-ConVQA is manually generated by annotators. Finally, back-translation, translating to another language and back, is a high-coverage although low-quality approach to text augmentation. It was used during training and shown to improve NLP models (Sennrich et al., 2016), but was not considered in VQA. We use English-German translations.

Models The VQA-CP models we consider are RUBi (Cadene et al., 2019), LMH (Clark et al., 2019) and CSS (Chen et al., 2020). The VQA models we consider are BUTD (Anderson et al., 2018), BAN (Kim et al., 2018), Pythia (Jiang et al., 2018) and VisualBERT (Li et al., 2019). For VisDial we use FGA (Schwartz et al., 2019) and VisDialBERT (Murahari et al., 2020). We trained all the models using their official implementations.

3.2 Results

Table 2 presents our main results. RAD values for all of our augmentations are substantially lower than those of the alternatives, supporting the value of our focused intervention approach for measuring robustness. The high RAD values for BT and Reph might indicate that VQA models are indeed robust to linguistic variation, as long as the answer does not change. Interestingly, our augmentations also reveal that VQA-CP models are less robust than VQA models. This suggests that despite the attempt to design more robust models, VQA-CP models still overfit their training data.

In VQA-CP, RUBi has the lowest accuracy performance in terms of its validation accuracy, even though it is more robust to augmentations compared with LMH and CSS. For VQA models, in contrast, BUTD has the lowest RAD scores on our augmentations and the lowest accuracy. VisualBERT, the only model that utilizes contextual word embeddings, demonstrates the highest robustness among the VQA models.

Finally, while both VisDial models have similar accuracy, they have significantly different RAD scores on our augmentations. Specifically, VisDialBERT performs better than FGA on Y/N ← C

augmentations. This is another indication of the value of our approach as it can help distinguish between two seemingly very similar models.

Complementary to the RAD values in Table 2 we also provide accuracies on original questions in Table 3. Note that across all the original questions, except ConVQA questions, RUBi has the lowest accuracy while CSS has the highest accuracy. This trend is reversed when looking at RAD scores - CSS has the lowest score while RUBi has the highest score. This emphasizes the importance of RAD as a complementary metric, since considering only accuracy in this case would be misleading. Namely, RAD provides additional critical information for model selection.

4 Measuring Generalization with RAD

To establish the connection between RAD and generalization, we design experiments to demonstrate RAD's added value in predicting model accuracy on unseen modified examples. Concretely, we generate 45 BUTD (VQA) and LMH (VQA-CP) instances, differing by the distribution of question types observed during training (for each model instance we drop between 10% and 99% of each of the question types "what color", "how many" and "what kind" from its training data; see Appendix E for exact implementation details). For each of the above models we calculate RAD values and accuracies in the following manner.

We split the validation set into two parts: $\mathcal{D}$ (features) and $\mathcal{T}$ (target). Consider a pool of four original question sets that are taken from their corresponding modifications: Y/N ← C, Y/N ← HM, Y/N ← WK, Reph. Then we have four possible configurations in which $\mathcal{D}$ is three sets from the pool and $\mathcal{T}$ is the remaining set. For each model and for each configuration, we compute model accuracy on $\mathcal{D}$ ($\mathrm{Accuracy}(\mathcal{D})$) and on the modifications of questions in $\mathcal{T}$ (the predicted variable $y(\mathcal{T}) = \mathrm{Accuracy}(\mathcal{T}')$) which are modified with the target augmentation of the experiment. We also compute the RAD values of the model on the modified questions in $\mathcal{D}$ which are generated using the other three augmentations ($\mathrm{RAD}(\mathcal{D}, \mathcal{D}')$, and $\mathrm{RAD}(\mathcal{D}', \mathcal{D})$). Then, we train a linear regression model using $\mathrm{Accuracy}(\mathcal{D})$, $\mathrm{RAD}(\mathcal{D}, \mathcal{D}')$, and $\mathrm{RAD}(\mathcal{D}', \mathcal{D})$, trying to predict $y(\mathcal{T})$. We perform this experiment four times, each using a different configuration (different augmentation type as our target), and average across the configurations.

Features\Model	R^2
	LMH
$\mathrm{Accuracy}(\mathcal{D})$, $\mathrm{RAD}(\mathcal{D}, \mathcal{D}')$, $\mathrm{RAD}(\mathcal{D}', \mathcal{D})$	0.917 ± 0.117
$\mathrm{Accuracy}(\mathcal{D})$	0.829 ± 0.237
$\mathrm{RAD}(\mathcal{D}, \mathcal{D}')$	0.899 ± 0.133
$\mathrm{RAD}(\mathcal{D}', \mathcal{D})$	0.849 ± 0.213

Table 4: Linear regression experiments, predicting accuracy performance on unseen augmentation types.

Results Table 4 presents the average R^2 values and standard deviations over the four experiments. RAD improves the R^2 when used alongside the validation accuracy. Interestingly, a model's accuracy on one set of augmentations does not always generalize to other, unseen augmentations. Only when adding RAD to the regression model are we able to identify a robust model. Notably, for LMH the usefulness of RAD is significant, as it improves the R^2 by 11%. It also predicts performance better than validation accuracy when used without it in the regression. The standard deviations further confirm that the above claims hold over all configurations. These observations hold when running the same experiment with respect to the BUTD model, however, the improvements are smaller since the regression task is much easier with respect to this model (R^2 of 0.995 with all features).

5 Conclusion

We proposed RAD, a new measure that penalizes models for inconsistent predictions over data augmentations. We used it to show that state-of-the-art VQA models fail on CADs that we would expect them to properly address. Moreover, we have demonstrated the value of our CADs by showing that alternative augmentation methods cannot identify robustness differences as effectively. Finally, we have shown that RAD is predictive of generalization to unseen augmentation types.

We believe that the RAD measure brings substantial value to model evaluation and consequently to model selection. It encourages the good practice of testing on augmented data, which was shown to uncover considerable model weaknesses in NLP (Ribeiro et al., 2020). Further, given visual augmentations, which we plan to explore in future work, or linguistic augmentations, RAD is applicable to any classification task, providing researchers with meaningful indications of robustness.

References

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. 2020. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *CVPR*.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. 2019. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*.

Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. 2020. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *EMNLP*.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2020. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *CVPR*.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2020. Causalm: Causal model explanation through counterfactual language models. *arXiv preprint arXiv:2005.13407*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *AAAI*.

Itai Gat, Idan Schwartz, Alexander Schwing, and Tamir Hazan. 2020. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. In *NeurIPS*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*.

Kushal Kafle, Robik Shrestha, and Christopher Kanan. 2019. Challenges and prospects in vision and language research. *Frontiers in Artificial Intelligence*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NeurIPS*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *NeurIPS*.

Vishvak Murahari, Dhruv Batra, Devi Parikh, and Abhishek Das. 2020. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. In *ECCV*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NuerIPS*.

Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. 2019. Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. In *EMNLP*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *ACL*.

Idan Schwartz, Seunghak Yu, Tamir Hazan, and Alexander G Schwing. 2019. Factor graph attention. In *CVPR*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In *CVPR*.

Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2020. Mmf: A multimodal framework for vision and language research. https://github.com/facebookresearch/mmf.

Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. 2020. On the value of out-of-distribution testing: An example of goodhart's law. *NeurIPS*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *CVPR*.

A Dataset Statistics

Please see Table 5 for the number of examples in each dataset that we use (VQA, VQA-CP and VisDial). We also report the number of augmentations we produce for each of our three augmentation types (Y/N ← C, Y/N ← HM and Y/N ← WK), alongside previous augmentation approaches used in our experiments (BT, Reph, L-ConVQA and CS-ConVQA).

B Our Augmentations

We describe the manual steps required to meet the desired standard for each augmentation type. For Y/N ← C, we filter out questions that start with "What color is the". For Y/N ← HM, we use questions that starts with "How many". For Y/N ← WK, we consider questions that match the pattern "What kind of <*S*> is this? <*O1*>". Table 6 presents several realizations of the templates we define (see Section 2.2 for a discussion of these templates).

In Y/N ← HM, we ensure that when the answer is '1', we use "Is there ..." instead of "Are there ...". We also ensure that the subsequent word to "How many" is a noun. We verify it is a noun using the part-of-speech tagger available through the spaCy library (Honnibal et al., 2020).

We allow the generation of both 'yes' and 'no' answers. Creating a modified question that is answered with a 'yes' requires a simple permutation of words in the original question-answer pair, e.g., for Y/N ← C, take "<*C1*>" = "<*C2*>" (see Table 1). Similarly, to generate a question that should be answered with a 'no', we repeat the above process and change "<*C2*>". In this case, we randomly pick an answer and replace it with the original answer with probability weighted with respect to the frequency in the data, among the pool of possible answers for the given augmentation type. When generating a new question, we first randomly decide whether to generate a 'yes' or 'no' question (with a probability of 0.5 for each). Then, for example, if we choose to generate a 'no', and "<*C1*>" = "red", we have a 63% chance of having "<*C2*>" = "blue".

C URLs of Data and Code

Data We consider three VQA datasets:

- The VQAv2 dataset (Goyal et al., 2017): https://visualqa.org/.
- The VQA-CPv2 dataset (Agrawal et al., 2018): https://www.cc.gatech.edu/grads/a/aagrawal307/vqa-cp/.
- The VisDial dataset (Das et al., 2017): https://visualdialog.org/

We also consider three previous augmentation methods:

- VQA-Rephrasings (Shah et al., 2019): https://facebookresearch.github.io/VQA-Rephrasings/.
- ConVQA (Ray et al., 2019): https://arijitray1993.github.io/ConVQA/.
- Back-translations (Sennrich et al., 2016). We have generated these utilizing the transformers library (Wolf et al., 2020), https://github.com/huggingface/transformers. Specifically, we used two pre-trained translation models, English to German, and German to English: https://huggingface.co/Helsinki-NLP/opus-mt-en-de, https://huggingface.co/Helsinki-NLP/opus-mt-de-en.

Models We consider nine models, where each model's code was taken from the official implementation. All implementations are via PyTorch (Paszke et al., 2019).

The three VQA-CPv2 models:

- RUBi (Cadene et al., 2019): https://github.com/cdancette/rubi.bootstrap.pytorch.
- LMH (Clark et al., 2019): https://github.com/chrisc36/bottom-up-attention-vqa.
- CSS (Chen et al., 2020): https://github.com/yanxinzju/CSS-VQA.

The four VQAv2 models:

- BUTD (Anderson et al., 2018): https://github.com/hengyuan-hu/bottom-up-attention-vqa.
- BAN (Kim et al., 2018): https://github.com/jnhwkim/ban-vqa.
- Pythia (Jiang et al., 2018): Using the implementation in the MMF library (Singh et al., 2020), https://github.com/facebookresearch/mmf.
- VisualBERT (Li et al., 2019): Using the implementation in the MMF library.

And the two VisDial models:

Dataset	Augmentation Count							Validation Count
	Y/N ← C	Y/N ← HM	Y/N ← WK	BT	Reph	L-ConVQA	CS-ConVQA	
VQA-CP	12,910	13,437	1,346	149,329	39,936	127,924	423	219,928
VQA	12,835	10,233	1,654	138,043	121,512	127,924	1,365	214,354
VisDial	516	130	-	1,136	-	-	-	20,640

Table 5: Number of examples in each of the datasets we use.

Yes/No ← Colors	Yes/No ← How Many	Yes/No ← What Kind
What color is the cat? White Is the color of the cat white? Yes	How many athletes are on the field? 5 Are there five athletes on the field? Yes	What kind of food is this? Breakfast Is this food breakfast? Yes
What color is the court? Green Is the color of the court green? Yes	How many dogs are in the picture? 3 Are there two dogs in the picture? No	What kind of event is this? Skiing Is this a skiing event? Yes
What color is the vase? Blue Is the color of the vase red? No	How many giraffes are walking around? 2 Are there four giraffes walking around? No	What kind of animal is this? Cow Is this animal an elephant? No
What color is the man's hat? Red Is the color of the man's hat red? Yes	How many cakes are on the table? 0 Is there one cake on the table? No	What kind of building is this? Church Is this building a church? Yes
What color is the sky? Blue Is the color of the sky blue? Yes	How many dogs? 1 Are there zero dogs? No	What kind of floor is this? Wood Is this a wood floor? Yes

Table 6: Some realizations of our templates (defined in Table 1). The black text (top) is the original question-answer pair and the blue text (bottom) is the corresponding augmented question-answer pair.

- FGA (Schwartz et al., 2019): `https://github.com/idansc/fga`.
- VisDialBERT (Murahari et al., 2020): `https://github.com/vmurahari3/visdial-bert`.

D Model Settings

We have trained the VQAv2 and the VQA-CPv2 models that we use, as pre-trained weights were not available for our requirements. For our evaluations, we require a model that is trained solely on the VQAv2 train set, such that we match the VQA-CPv2 settings, where there are only two sets, train and validation. In contrast, pre-trained models that are built for VQAv2 are trained on the VQAv2 training set and on the VQAv2 validation set together, as the dataset contains a third development set that is commonly used for validation.

We have trained six VQA models using the default hyper-parameters from their official implementations (URLs in Appendix C): RUBi, LMH, CSS, BUTD, BAN and Pythia. We trained the above models on a single Nvidia GeForce RTX 2080 Ti GPU, when the training time for each of the models was less than 12 hours. In addition, inference in this setting took less than an hour for all models.

The VisualBERT model is more computationally intensive, and we had to reduce the default batch size from 480 to 54 to fit it on our resources. Using three Nvidia GeForce RTX 2080 Ti GPUs for VisualBERT, training took 36 hours and inference took 4 hours.

For the VisDial models, FGA, and VisDialBERT, we have downloaded the pre-trained weights and used them solely for inference. On a single Nvidia GeForce RTX 2080 Ti GPU, inference took 15 minutes for FGA, and 8 hours for VisDialBERT.

All the models we consider have less than 200M parameters.

When accuracies are reported on VQAv2 and on VQA-CP (Tables 2 and 3) we use the VQA-accuracy metric (Antol et al., 2015). For VisDial we use the standard accuracy metric (denoted originally as R@1).

E Regression Experiments

We generate 45 BUTD (VQA) instances and 45 LMH (VQA-CP) instances. To generate different model instances, we create 45 new training sets by removing examples from the original train set. For each of the three question types, "what color", "how many" and "what kind", we remove the following 15 percentage values of examples from the original train set: [10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 92%, 95%, 96%, 97%, 98%, 99%], resulting in 45 new training sets. Then, each model instance is created by training on one of the 45 training sets.

We split the validation set into two parts: $\mathcal{D}$ and $\mathcal{T}$. $\mathcal{D}$ is used to calculate the features in our linear

regression model. We denote with $\mathcal{D}'_1$ the questions in $\mathcal{D}$ that can be modified using the Y/N $\leftarrow$ C augmentation, after these questions were modified. Similarly, we define $\mathcal{D}'_2$, $\mathcal{D}'_3$, and $\mathcal{D}'_4$ for Y/N $\leftarrow$ HM, Y/N $\leftarrow$ WK, and Reph, respectively.

We average the R^2 of four linear regression experiments, when in each experiment we set a different i ($i \in \{1, 2, 3, 4\}$) for which $\mathcal{T} = \mathcal{D}'_i$ and use the remaining three templates to calculate our features. We denote the regression features with $x_1 = \mathrm{Accuracy}(\mathcal{D})$, $x_2 = \mathrm{RAD}(\mathcal{D}, \mathcal{D}')$, and $x_3 = \mathrm{RAD}(\mathcal{D}', \mathcal{D})$, where $\mathrm{RAD}(\mathcal{D}, \mathcal{D}')$ and $\mathrm{RAD}(\mathcal{D}', \mathcal{D})$ are computed with respect to the three other templates ($j \in \{1, 2, 3, 4\}, j \neq i$). The predicted label is $y(\mathcal{T}) = \mathrm{Accuracy}(\mathcal{T})$.

Thus the equation for our regression is:

$$y(\mathcal{T}) = b_1 x_1 + b_2 x_2 + b_3 x_3 + \epsilon \ .$$

We also perform three regression experiment for each feature alone:

$$y(\mathcal{T}) = b x_k + \epsilon, \quad k = 1, 2, 3 \ ,$$

and compare the results of these experiments in Table 4.

How Helpful is Inverse Reinforcement Learning for Table-to-Text Generation?

Sayan Ghosh[†*] **Zheng Qi**[‡*] **Snigdha Chaturvedi**[†] **Shashank Srivastava**[†]
† UNC Chapel Hill
‡ University of Pennsylvania
{sayghosh, snigdha, ssrivastava}@cs.unc.edu
issacqzh@seas.upenn.edu

Abstract

Existing approaches for the Table-to-Text task suffer from issues such as missing information, hallucination and repetition. Many approaches to this problem use Reinforcement Learning (RL), which maximizes a single manually defined reward, such as BLEU. In this work, we instead pose the Table-to-Text task as Inverse Reinforcement Learning (IRL) problem. We explore using multiple interpretable unsupervised reward components that are combined linearly to form a composite reward function. The composite reward function and the description generator are learned jointly. We find that IRL outperforms strong RL baselines marginally. We further study the generalization of learned IRL rewards in scenarios involving domain adaptation. Our experiments reveal significant challenges in using IRL for this task.

1 Introduction

Table-to-Text generation focuses on explaining tabular data in natural language. This is increasingly relevant due to the vast amounts of tabular data created in domains including e-commerce, healthcare and industry (for example, infoboxes in Wikipedia, tabular product descriptions in online shopping sites, etc.). Table-to-Text can make data easily accessible to non-experts and can automate certain pipelines like auto-generation of product descriptions. Traditional methods approached the general problem of converting structured data to text using slot-filling techniques (Kukich, 1983; Reiter and Dale, 2000; McKeown, 1992; Cawsey et al., 1997; Konstas and Lapata, 2013; Flanigan et al., 2016). While recent advances in data-to-text generation using neural networks (Sutskever et al., 2011; Mei et al., 2015; Gardent et al., 2017; Wiseman et al., 2017; Song et al., 2018; Zhao et al., 2020) have led to improved fluency, current systems still suffer from issues such as lack of coverage (where the generated text misses information present in the source), repetition (where the generated text repeats information) and hallucination (where the generated text asserts information not present in the source)(Lee et al., 2019). A significant reason for these issues is that models often lack explicit inductive biases to avoid these problems. Most extant approaches utilize Reinforcement Learning-based (RL) training, using a single reward (such as BLEU or task-specific rewards) that optimizes for a specific aspect. For example, Liu et al. (2019) and Nishino et al. (2020) use domain-specific rewards to improve the accuracy of medical report generation.

However, defining a single reward that addresses all of the above-described issues is difficult. To use multiple reward components with RL, one has to manually find an optimal set of weights of each component either through a trial-and-error approach or expensive grid search which gets infeasible as the number of such reward components increases. Inverse Reinforcement Learning (Abbeel and Ng, 2004; Ratliff et al., 2006; Ziebart et al., 2008) can be a natural approach for this task since it can learn an underlying composite reward function from labeled examples incorporating multiple rewards. Motivated by existing applications of IRL in other domains and tasks (Finn et al., 2016; Fu et al., 2017; Shi et al., 2018), we explore its utility for Table-to-Text generation. We diverge from previous work on IRL in designing a set of intuitive and interpretable reward components that are linearly combined to get the reward function. Figure 1 illustrates the overall idea of this work. We learn a "Description Generator" (also referred as policy later) to generate descriptions given a table. The IRL framework includes "Reward Approximator" that leverages the "expert" or the ground-truth de-

[*] Authors contributed equally.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 71–79
August 1–6, 2021. ©2021 Association for Computational Linguistics

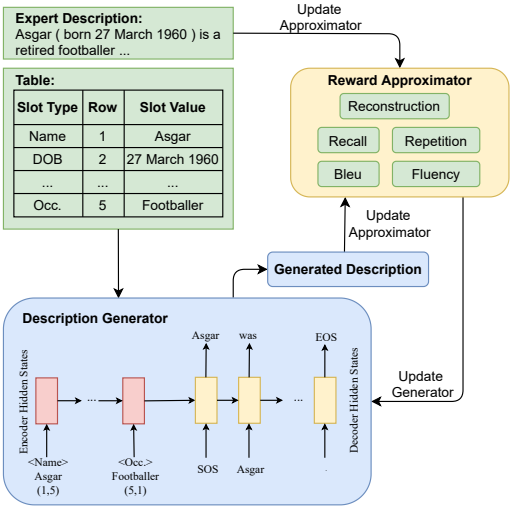


Figure 1: We frame the Table-to-Text task under Inverse Reinforcement Learning framework using multiple reward components

scriptions corresponding to tables to jointly learn the underlying composite reward function combining multiple reward components such as "Recall", "Fluency", etc. This composite reward function quantifies the quality of the generated descriptions. We see IRL performs at par with RL baselines. For investigating when IRL helps and when it does not, we conduct experiments to evaluate generalization capabilities of IRL in limited data setting and identify challenges involved in IRL training. Our contributions are:

- We formulate a set of interpretable reward components and learn the composite linear reward function in a data-driven manner for Table-to-Text generation[1].
- We study the utility of IRL for Table-to-Text generation.

2 Method

The training data for this task consists of pairs of tables and corresponding natural language descriptions, as shown in Figure 1. A table T is a sequence of tuples of slot types (e.g. "Name") and slot values (e.g. "Asgar") and let D denote the expert description. We formulate the "Table-to-Text" task as generating D from source table T. In the rest of this section, we first explain how to formulate Table-to-Text under the IRL framework, followed by the formulation of the reward components and a brief description of the text generation network that is at the core of our method.

2.1 Table-to-Text as IRL

We pose Table-to-Text under the IRL framework where we aim to jointly learn a policy for generating description from the table and the underlying composite reward function. At the core of our approach, we have a neural description generator that we adapt from Wang et al. (2018). The description generator is first trained using maximum likelihood estimation (MLE) followed by fine-tuning it using Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) (Ziebart et al., 2008). Under the MaxEnt IRL framework, we iteratively perform two steps: (1) approximate the underlying composite reward function by leveraging the expert descriptions and the current policy for description generation; (2) Using the updated reward function, we update the current policy for description generation using RL. In this work, we model the composite reward $R_\phi(D)$ as a linear combination of multiple reward components.

$$R_\phi(D) = \sum_{t=1}^{\tau} \phi^\intercal C^t \tag{1}$$

where ϕ is a weight vector, C^t is the vector of reward component values at step t in a generated description and τ denotes total steps.

Following the MaxEnt IRL paradigm, we assume the expert descriptions come from a log-linear distribution ($p_\phi(D)$) on reward values. The objective of the reward approximator ($J_r(\phi)$) is to maximize the likelihood of the expert descriptions. The partition function for this distribution ($p_\phi(D)$) is approximated by using importance sampling from the learned description generation policy. For sake of brevity, we skip the mathematical derivation here. Please refer to Appendix A.1 for detailed derivation. We draw N expert descriptions and M descriptions from the learned policy. The gradient of the objective ($J_r(\phi)$) w.r.t. reward function parameters ϕ is then the difference between the expected expert reward and expected reward obtained by the policy (Ziebart et al., 2008):

$$\nabla_\phi J_r(\phi) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\phi R_\phi(D_i) - \frac{1}{\sum_j \beta_j}\sum_{j=1}^{M} \beta_j \nabla_\phi R_\phi(D'_j) \tag{2}$$

where D_i and D'_j are drawn from the training data and the learned policy respectively and β's are importance sampling weights.

[1]Code and dataset splits for the paper are provided in https://github.com/issacqzh/IRL_Table2Text

The linear functional form of the reward simplifies individual weight updates as a simple difference of the expected expert and the expected roll out reward component from policy. Weight update for component c is:

$$\nabla_\phi J_r(\phi)_c = \frac{1}{N}\sum_{i=1}^{N} c_i - \frac{1}{\sum_j \beta_j}\sum_{j=1}^{M} \beta_j c'_j \quad (3)$$

where c_i is total value of reward component over all steps for i^{th} expert description and c'_j is total value of reward component over all steps for j^{th} generated description. To stabilize training when learning the policy for description generation we mix in weighted MLE loss with the policy gradient loss before backpropagation. Please refer to supplementary material (Appendix A.5) for model training details.

2.2 Reward Components

We aim to find a reward function that can combine multiple characteristics present in a good description such as faithfulness to the table and fluency. To encourage faithfulness, we use *recall* and *reconstruction* as reward components, while to characterize grammatical correctness and fluency we use *repetition* and *perplexity*. We also consider *BLEU* score as a reward component. BLEU is a supervised reward component as it requires ground-truth descriptions for its computation. However, all other reward components are unsupervised.

- **Recall:** Fraction of slot values in the table mentioned in the description.
- **Reconstruction:** We use QA models to extract answers from the description against a few "extractor" slot types (for example, "What is the name of the person in the description?" is used as a question for the slot type "Name_ID"). Details about other extractor slot types are provided in Appendix A.3. Reconstruction score is the average of lexical overlap scores between predicted and true slot values, corresponding to the extractor slot types present in the table.
- **Repetition:** Fraction of unique trigrams in the description.
- **Perplexity:** This is the normalized perplexity of the description calculated using GPT-2 model (Radford et al., 2019).
- **BLEU:** This is the BLEU score (Papineni et al., 2002) of the description.

Additional details on implementation of reward components are in Appendix A.3.

3 Experiments and Results

In this section we describe our experiments and their results in detail.

3.1 Data and Metrics

Wang et al. (2018) proposed a dataset of tables and their corresponding descriptions related to people and animals from Wikipedia. However, the original released dataset is noisy (many descriptions have low precision/recall, most examples have very few distinct slot types, etc.). For our experiments, we filtered this dataset to get a smaller high-quality dataset of 4623 examples using the following criteria : (1) Recall (defined in §2.2) of 1.0 (2) High precision (fraction of entities in the description mentioned in the table) greater than 0.7 (3) number of distinct slot types greater than 6. We split the entire dataset as 80%, 10% and 10% for training, validation and testing respectively. Details of the dataset are provided in Appendix A.2. To aid reproducibility we make the data splits used by us publicly available[2].

For evaluation, we report BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) along with their harmonic mean (called F1 hereon). Additionally, we report the mean reward value for Recall and Perplexity as proxies for faithfulness and fluency of generated descriptions respectively.

MODEL	B	R	F1	REC.	PPL
MLE	23.78	42.11	30.40	0.82	-73.38
RL with					
B	28.03	43.75	34.17	0.87	-42.17
Rec., B	28.02	43.77	34.16	0.88	-39.43
Rec., B, PPL	28.22	43.12	34.11	0.89	-43.91
All	28.21	43.23	34.14	0.89	-40.77
IRL with					
Rec., B	27.96	43.52	34.04	0.88	-40.27
Rec., B, PPL	28.25	**43.81**	34.35	0.89	-40.19
All	**28.42**	43.19	34.28	0.89	-40.11
IRL (using multipliers) with					
Rec., B	28.41	43.53	**34.38**	0.89	**-38.53**
Rec., B, PPL	28.16	43.35	34.14	**0.90**	-40.86

Table 1: Test set performance for various models measured using BLEU (B), ROUGE(R), F1, Recall(Rec.) and Perplexity (PPL). Using IRL instead of RL gives marginal improvement in performance

3.2 Automatic Evaluation

Table 1 shows the performance of models trained using maximum likelihood estimation (MLE), RL

[2]Data splits are provided in https://github.com/issacqzh/IRL_Table2Text

and IRL. For RL and IRL we report results with various sets of reward components. When using multiple reward components with RL we consider the total reward as the uniformly weighted sum of each component. We note that while IRL variants achieve higher performance than RL methods for all metrics, the gain in performance is marginal.

In Table 1 we choose the best model for each setting based on the performance on the validation split. For the best IRL (All) model, we find the learned weights for repetition, recall, BLEU, reconstruction and perplexity are 0.02, 0.12, 0.65, 0.05 and 0.15 respectively. However, we noticed that the weights of the IRL reward components failed to converge in our training runs. This is a consequence of the fact that reward components such as BLEU achieve their maximum value for the ground-truth description, and the value quickly drops as descriptions diverge from the ground truth description. Thus the gap between the expert value and the value achieved by the model for BLEU is always large, hindering the convergence of weights in IRL (Eqn 3). This results in a peaked distribution of weights where the model tends to favor the BLEU reward component excessively. We attempt scaling down the expert BLEU reward values by using multiplier. We dynamically update the multiplier using an adaptive binary search method (refer to Appendix A.4 for details) to induce convergence in weights. We observe that the multiplier acts as a "regularizer" in learning a more balanced weight for the reward components considered. For example, when we train IRL with BLEU, recall and perplexity without using multiplier, the learned weights of the components are 0.72, 0.15 and 0.13 respectively. On using multipliers for IRL training, the learned weights for BLEU, recall and perplexity are 0.45, 0.31 and 0.24 respectively. The model variants using multipliers get the best F1 score as seen in the second last row of Table 1.

We also find that having more reward components does not help IRL improve significantly. We note that IRL using all reward components gets the best BLEU but suffers a marginal drop in ROUGE.

3.3 Domain adaptation

To evaluate if rewards learned using IRL generalize better to unseen data distributions, we evaluate it for scenarios involving domain adaptation. For this, we divide the dataset into disjoint subsets of categories involving people in sports, academia, art, etc. (category details in Appendix A.2). Each category has different table schemas. We train RL and IRL models on one category and test them on a different category. Since training on a single category limits the amount of labelled data, we consider training with unsupervised rewards that do not rely on the ground truth. Table 2 shows the F1 results when using IRL and RL with recall, perplexity and reconstruction. For each training category, we show results of the test category with the highest absolute value of relative change in F1. We notice mixed results. For instance, when training on the "Sports" domain, IRL's performance is much worse than RL. This may be because slot types with high frequency in the "Sports" category are significantly different from all other categories. Thus, IRL may be susceptible to learning a reward function that overfits the domain and actually generalizes worse than a fixed reward function. However, in several cases IRL leads to big improvements in performance (e.g. when training on Politics, Law, and Military) indicating the promise of this method in limited data settings.

TRAIN CAT.	TEST CAT.	RL	IRL
Politics	Sports	18.59	**21.04**
Law	Academia	28.54	**31.17**
Military	Politics	30.07	**32.01**
Art	Academia	**32.78**	31.37
Academia	Sports	**21.25**	20.67
Sports	Academia	**24.43**	22.25

Table 2: F1 scores on using IRL and RL for domain adaptation. IRL leads to higher F1 scores in a few settings indicating its usefulness for domain adaptation. However, IRL performs worse than RL when trained on domains which have significantly different slot types with high frequency (e.g. "Sports").

4 Discussion

We highlight some challenges with IRL training that potentially hinder IRL to get significantly better than RL baselines. Further, we discuss qualitative differences between RL and IRL models.

4.1 Challenges in IRL training

Importance of reward components: During training, for most reward components, their values for expert and generated descriptions are close. However, the values of BLEU for generated descriptions are quite smaller than the BLEU value for expert descriptions. This shadows the contribution of other reward components irrespective of

the weights assigned to them. Since BLEU optimizes for n-gram overlap with the expert text, it is undesirable to drop this component as it leads to text degeneration. As described in Section 3.2, we use adaptive multipliers to alleviate its dominance. However, its effect is limited and the method does not correspond to optimizing a fixed objective.

Unstable training: To stabilize training, we mix the weighted MLE loss (cross-entropy loss) and the policy gradient objective. However, these losses can differ largely in scale. Having a larger weight to MLE loss diminishes the contribution of reward components, while larger weight to policy gradient leads to degeneration.

These observations indicate the need for future research on training paradigms and better-designed reward components to address these challenges.

4.2 Qualitative analysis

Using only BLEU as a reward leads to generated descriptions that fit a general template resembling descriptions from the most common category ("Sports"). Including other reward components helps the model avoid this behavior. We still observe hallucination from both IRL and RL fine-tuned models. However, hallucinated information generated from IRL fine-tuned models often matches the overall theme (for example, it generates incorrect football league names but gets the name of the club mentioned in the table correct). Appendix A.7 shows an example of description generated by IRL (All) model.

5 Conclusion

We present an approach using IRL for Table-to-Text generation using a set of interpretable reward components. While the approach outperforms RL, improvements are marginal, and we identify several challenges. In particular, using metrics like BLEU as reward components is problematic, since they affect weight convergence for IRL. Based on our study, the application of IRL for Table-to-Text generation would broadly benefit from designing better-calibrated reward components and improvements in training paradigms. We hope our exploration encourages the community to engage in interesting directions of future work.

References

Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the twenty-first international conference on Machine learning, page 1.

Alison J Cawsey, Bonnie L Webber, and Ray B Jones. 1997. Natural language generation in health care.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. arXiv preprint arXiv:1611.03852.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime G Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 731–739.

Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. arXiv preprint arXiv:1710.11248.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In 55th annual meeting of the Association for Computational Linguistics (ACL).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. Journal of Artificial Intelligence Research, 48:305–346.

Karen Kukich. 1983. Design of a knowledge-based report generator. In 21st Annual Meeting of the Association for Computational Linguistics, pages 145–150.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pages 74–81.

Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, Willie Boag, Wei-Hung Weng, Peter Szolovits, and Marzyeh Ghassemi. 2019. Clinically accurate chest x-ray report generation. In Machine Learning for Healthcare Conference, pages 249–269. PMLR.

Kathleen McKeown. 1992. Text generation. Cambridge University Press.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. arXiv preprint arXiv:1509.00838.

Toru Nishino, Ryota Ozaki, Yohei Momoki, Tomoki Taniguchi, Ryuji Kano, Norihisa Nakano, Yuki Tagawa, Motoki Taniguchi, Tomoko Ohkuma, and Keigo Nakamura. 2020. Reinforcement learning with imbalanced dataset for data-to-text medical report generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 2223–2236.

Travis E Oliphant. 2006. A guide to NumPy, volume 1. Trelgol Publishing USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. 2006. Maximum margin planning. In Proceedings of the 23rd international conference on Machine learning, pages 729–736.

Ehud Reiter and Robert Dale. 2000. Building Natural Language Generation Systems. Studies in Natural Language Processing. Cambridge University Press.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward diverse text generation with inverse reinforcement learning. arXiv preprint arXiv:1804.11258.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. arXiv preprint arXiv:1805.02473.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In ICML.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In Proceedings of the 11th International Conference on Natural Language Generation, pages 10–21. Association for Computational Linguistics.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. arXiv preprint arXiv:1707.08052.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2481–2491, Online. Association for Computational Linguistics.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. In Aaai, volume 8, pages 1433–1438. Chicago, IL, USA.

A Appendices

A.1 Derivation of gradient for MaxEnt IRL

We show the detailed mathematical steps to approximate the gradient calculation under the MaxEnt IRL framework for Table-to-Text generation.

We assume that expert descriptions are drawn from a distribution $p_\phi(D)$.

$$p_\phi(D) = \frac{1}{Z} exp(R_\phi(D)) \text{ and } Z = \int_D exp(R_\phi(D)) \quad (4)$$

where the reward function, $R_\phi(D)$ has parameters ϕ, and Z is the partition function. The total reward

of a description is sum of rewards at each step. Let $q_\theta(D)$ be the policy for description generation. We maximise the log-likelihood of the samples in the training set (Equation 5).

$$J_r(\phi) = \frac{1}{N}\sum_{n=1}^{N} log(p_\phi(D_n)) = \frac{1}{N}\sum_{n=1}^{N} R_\phi(D_n) - logZ \tag{5}$$

The gradient w.r.t. reward parameters is given by

$$\begin{aligned}\nabla_\phi J_r(\phi) &= \frac{1}{N}\sum_{n} \nabla_\phi R_\phi(D_n) \\ &\quad - \frac{1}{Z}\int_D exp(R_\phi(D))\nabla_\phi R_\phi(D) dD \\ &= \mathbb{E}_{D\sim p_{data}} \nabla_\phi R_\phi(D) - \mathbb{E}_{D\sim p_{\phi(D)}} \nabla_\phi R_\phi(D)\end{aligned} \tag{6}$$

The partition function requires enumerating all possible descriptions which makes this intractable. This is tackled by approximating the partition function by sampling descriptions from the policy using importance sampling. The importance weight β_i for a generated description D_i is given by

$$\beta_i \propto \frac{exp(R_\phi(D_i))}{q_\theta(D_i)} \tag{7}$$

The gradient is now approximated as:

$$\nabla_\phi J_r(\phi) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\phi R_\phi(D_i) - \frac{1}{\sum_j \beta_j}\sum_{j=1}^{M} \beta_j \nabla_\phi R_\phi(D'_j) \tag{8}$$

where D_i and D'_j are drawn from training data and $q_\theta(D)$ respectively.

A.2 Dataset statistics

We split the entire dataset as 80%, 10% and 10% for training, validation and testing respectively. Table 3 shows the statistics for our dataset.

Table 4 shows the various disjoint category splits of our data.

A.3 Detailed description of some reward components

- **Reconstruction:** We use Question Answering models to extract answers from the description corresponding to few slot types. For example, to extract the name from the description we ask a question "What is the name of the person?". The questions corresponding to each slot type is pre-determined. We extract values for four most common slot types occurring in the dataset – "name", "place of birth", "place of death" and "country". We will refer to these slots as "extraction slot types". The questions for these extractor slot types are "What is the name of the person in the description?", "What is the place of birth of the person in the description?", "What is the place of death of the person in the description?" and "Which country does the person in the description belong to?" respectively. All extraction slot types are not present in every table of the dataset (example, "place of death" is not present for a living sportsperson). Following SQUAD-like (Rajpurkar et al., 2018) formalisation, for each slot-type we train a BERT-based (Devlin et al., 2019) model to get the answer from the description given the question. We calculate overlap score of predicted answer with the correct answer (slot value from table). The final reconstruction score is the arithmetic mean of answer overlap scores corresponding to the extractor slot types present in the table.
- **Perplexity:** This is the negative perplexity of the description. We further normalize it by using

$$\frac{\text{Perplexity} - \text{Perplexity}_{low}}{\text{Perplexity}_{high} - \text{Perplexity}_{low}} \tag{9}$$

where Perplexity_{high} and Perplexity_{low} are the maximum and minimum perplexity of expert texts and texts generated by pretrained MLE model respectively.

A.4 Learning Multiplier for BLEU

Let us assume that after the i^{th} iteration of IRL, we have the multiplier value as m_i. Let b be the average BLEU score obtained by the model. For $(i+1)^{th}$ iteration we update the multiplier value as

$$m_{i+1} = \frac{m_i + b}{2} \tag{10}$$

In case the change in weight is less than 0.00001, we instead increase multiplier value by 0.1. The maximum of multiplier value is 1. We start with initial multiplier value (m_0) as 1.

A.5 Training details

Model parameters We follow the same training scheme and model parameters from Wang et al. (2018). Our model roughly has around 7.8M parameters. We perform MLE for 20 epochs. For RL finetuning we perform 100 epochs. For the IRL

TYPE	SIZE	REC.	PREC.	# SENT./TAB.	# SLOTS/SENT.	# SLOTS/TAB.	# W/SENT.	# W/TAB.
Train	3700	1.0	0.82	4.61	1.86	8.58	14.85	68.52
Val	461	1.0	0.82	4.67	1.86	8.70	15.10	70.54
Test	462	1.0	0.82	4.60	1.86	8.57	14.54	66.87
Total	4623	1.0	0.82	4.62	1.86	8.59	14.85	68.56

Table 3: Dataset statistics

Slot Type	Row	Slot Value
Name ID	1	William Duval (ice hockey)
Country of citizenship	2	Canada
Date of birth	3	August 3, 1877
Date of death	4	June 7, 1905
Sport	5	Ice hockey
Position played on team / Specialty	6	Defenceman
Place of birth	7	Ottawa

Reference:
William Duval (ice hockey) (August 3 1877 – June 7 1905) was a Canadian professional Ice hockey Defenceman who played for the Ottawa Hockey Club and the Pittsburgh Victorias in the late 1890s and early 1900s . born in Ottawa Canada Duval played intermediate hockey for the Ottawa Aberdeens and Ottawa Atlantic Railway teams before joining the Ottawa Hockey Club in the 1899 – 1900 season . he played two further seasons for Ottawa and was named captain prior to the 1902 season . duval died due to alcoholism on June 7 1905 . duval had previously worked for the Canada Atlantic Railway in Ottawa .

IRL All:
William Duval (ice hockey) (August 3 1877 - June 7 1905) was a Canada professional Ice hockey Defenceman who played eleven seasons in the National Hockey League of six . he was born in Ottawa Ontario Canada .

Figure 2: Example of generated description using IRL (All) model

CATEGORY	# SAMPLES
Academia	2152
Art	4736
Politics	2974
Sports	17434
Law	586
Military	4170
Unknown	14096
All	46148

Table 4: Data statistics for categories

model, we perform two weight updates followed by five RL epochs and this is repeated 20 times. For training we use Adam optimizer (Kingma and Ba, 2014). We choose the hyperparameters and best epoch for each model by obtaining results on the validation set using beam search with beam size of 3.

Hyper-parameter tuning We adapt the model and optimizer hyper-parameters from Wang et al. (2018). For choosing the weights for cross-entropy loss and policy gradient loss we tried combinations in the set 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 keeping sum of weights as 1. For IRL reward component weight updates we sample 500 descriptions from ground-truth and from the descriptions generated from the policy. We chose the size as 500 based on validation set performance. Based on the performance on the validation set we chose 0.9 as policy gradient loss weight and 0.1 for cross-entropy loss. This also helps to bring both the loss terms in same scale.

Software and hardware specifications All the models are coded using Pytorch 1.4.0[3] (Paszke et al., 2019) and related libraries like numpy (Oliphant, 2006), scipy (Virtanen et al., 2020) etc. We run all experiments on GeForce RTX 2080 GPU of size 12 GB. The system has 256 GB RAM and 40 CPU cores.

Time for training and inference It takes around 16 seconds for one epoch of MLE training while it takes close to 150 seconds for an epoch when using RL fine-tuning with all the reward components. The reward component weight approximation stage of IRL is very fast and takes less than a second generally.

A.6 Validation set results

Table 5 shows the results on validation set for the models in Table 1 of main paper.

A.7 Qualitative example

Table 2 shows an example of the output generated by the IRL (All) model along with the reference

[3] https://pytorch.org/

MODEL	B	R	F1	REC.	PPL
MLE	23.64	41.17	30.03	0.83	-74.75
RL with					
B	26.61	**42.12**	32.61	0.86	-39.89
Rec., B	26.88	42.05	32.79	0.87	**-34.06**
Rec., B, PPL	27.10	41.72	32.86	0.88	-43.47
All	26.87	41.87	32.73	0.88	-39.14
IRL with					
Rec., B	26.85	41.79	32.69	0.87	-37.40
Rec., B, PPL	27.09	42.10	**32.97**	0.88	-40.13
All	**27.23**	41.70	32.94	0.88	-39.87
IRL (using multipliers) with					
Rec., B	27.02	42.00	32.88	0.86	-34.83
Rec., B, PPL	27.02	41.67	32.78	**0.89**	-40.87

Table 5: Performance on the validation set for various models measured using BLEU (B), ROUGE(R), F1, Recall(Rec.) and Perplexity(PPL)

description.

Automatic Fake News Detection: Are Models Learning to Reason?

Casper Hansen[*]
University of Copenhagen
c.hansen@di.ku.dk

Christian Hansen[*]
University of Copenhagen
chrh@di.ku.dk

Lucas Chaves Lima
University of Copenhagen
lcl@di.ku.dk

Abstract

Most fact checking models for automatic fake news detection are based on reasoning: given a claim with associated evidence, the models aim to estimate the claim veracity based on the supporting or refuting content within the evidence. When these models perform well, it is generally assumed to be due to the models having learned to reason over the evidence with regards to the claim. In this paper, we investigate this assumption of reasoning, by exploring the relationship and importance of both claim and evidence. Surprisingly, we find on political fact checking datasets that most often the highest effectiveness is obtained by utilizing only the evidence, as the impact of including the claim is either negligible or harmful to the effectiveness. This highlights an important problem in what constitutes evidence in existing approaches for automatic fake news detection.

1 Introduction

Misinformation is spreading at increasing rates (Vosoughi et al., 2018), particularly online, and is considered a highly pressing issue by the World Economic Forum (Howell et al, 2013). To combat this problem, automatic fact checking, especially for estimating the veracity of potential fake news, have been extensively researched (Hassan et al., 2017; Hansen et al., 2019; Thorne and Vlachos, 2018; Elsayed et al., 2019; Allein et al., 2020; Popat et al., 2018; Augenstein et al., 2019). Given a claim, most fact checking systems are *evidence-based*, meaning they utilize external knowledge to determine the claim veracity. Such external knowledge may consist of previously fact checked claims (Shaar et al., 2020), but it typically consists of using the claim to query the web through a search API to retrieve relevant hits. While including the evidence in the model increases the effectiveness over using only the claim, existing work has not focused on the predictive power of isolated evidence, and hence whether it assists the model in enabling better reasoning.

In this work we investigate if fact checking models learn reasoning, i.e., provided a claim and associated evidence, whether the model determines the claim veracity by reasoning over the evidence. If the model learns reasoning, we would expect the following proposition to hold: *A model using both the claim and evidence should perform better on the task of fact checking compared to a model using only the claim or evidence*. If a model is only given the claim as input, it does not necessarily have the information needed to determine the veracity. Similarly, if the model is only given the evidence, the predictive signal must come from dataset bias or the differences in the evidence obtained from claims with varying veracity, as it otherwise corresponds to being able to provide an answer to an unknown question. In our experimental evaluation on two political fact checking datasets, across multiple types of claim and evidence representations, we find the evidence provides a very strong predictive signal independent of the claim, and that the best performance is most often obtained while entirely ignoring the claim. This highlights that fact checking models may not be learning to reason, but instead exploit an inherent signal in the evidence itself, which can be used to determine factuality independent of using the claim as part of the model input. This highlights an important problem in what constitutes evidence in existing approaches for automatic fake news detection. We make our code publicly available[1].

[*]Equal contribution.

[1]https://github.com/casperhansen/fake-news-reasoning

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 80–86
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Related Work

Automatic fact checking models include deep learning approaches, based on contextual and non-contextual embeddings, which encode the claim and evidence using RNNs or Transformers (Shaar et al., 2020; Elsayed et al., 2019; Allein et al., 2020; Popat et al., 2018; Augenstein et al., 2019; Hassan et al., 2017), and non-deep learning approaches (Wang, 2017; Pérez-Rosas et al., 2018), which uses hand-crafted features or bag-of-word representations as input to traditional machine learning classifiers such as random forests, SVM, and MLP (Mihalcea and Strapparava, 2009; Pérez-Rosas et al., 2018; Baly et al., 2018; Reddy et al., 2018).

Generally, models may learn to memorize artifact and biases rather than truly learning (Gururangan et al., 2018; Moosavi and Strube, 2017; Agrawal et al., 2016), e.g., from political individuals often leaning towards one side of the truth spectrum. Additionally, language models have been shown to implicitly store world knowledge (Roberts et al., 2020), which in principle could enhance the aforementioned biases. To this end, we design our experimental setup to include representative fact checking models of varying complexity (from simple term-frequency based representations to contextual embeddings), while always evaluating each trained model on multiple different datasets to determine generalizability.

3 Methods

Problem definition. In automatic fact checking of fake news we are provided with a dataset of $D = \{(c_1, e_1, y_1), ..., (c_n, e_n, y_n)\}$, where c_i corresponds to a textual claim, e_i is evidence used to support or refute the claim, and y_i is the associated truth label to be predicted based on the claim and evidence. Following current work on fact checking of fake news (Hassan et al., 2017; Thorne and Vlachos, 2018; Elsayed et al., 2019; Allein et al., 2020; Popat et al., 2018; Augenstein et al., 2019), we consider the evidence to be a list of top-10 search snippets as returned by Google search API when using the claim as the query. Note that while additional metadata may be available–such as speaker, checker, and tags–this work focuses specifically on whether models learn to reason based on the combination of claim and evidence, hence we keep the input representation to consist only of the latter.

Overview. In the following we describe the different models used for the experimental comparison (Section 4), which consists of models based on term frequency (term-frequency weighted bag-of-words (Salton and Buckley, 1988)), word embeddings (GloVe word embeddings (Pennington et al., 2014)), and contextual word embeddings (BERT (Devlin et al., 2019)). These representations are chosen as to include the typical representations most broadly used among past and current NLP models.

Term-frequency based Random Forest. We construct a term-frequency weighted bag-of-words representation per sample based on concatenating the text content of the claim and associated evidence snippets. We train a Random Forest (Breiman, 2001) as the classifier using the Gini impurity measure. In the setting of only using either the claim or evidence snippets as the input, only the relevant part is used for constructing the bag-of-words representation.

GloVe-based LSTM model. We adapt the model by Augenstein et al. (2019), which originally was proposed for multi-domain veracity prediction. Using a pretrained GloVe embedding (Pennington et al., 2014)[2], claim and snippet tokens are embedded into a joint space. We encode the claim and snippets using an attention-weighted bidirectional LSTM (Hochreiter and Schmidhuber, 1997):

$$h_{c_i} = \text{attn}\left(\text{BiLSTM}(c_i)\right) \quad (1)$$

$$h_{e_{i,j}} = \text{attn}\left(\text{BiLSTM}(e_{i,j})\right) \quad (2)$$

where attn$(\cdot)$ is a function that learns an attention score per element, which is normalized using a softmax, and returns a weighted sum. We combine the claim and snippet encodings using using the matching model by Mou et al. (2016) as:

$$s_{i,j} = \left[h_{c_i} \; ; \; h_{e_{i,j}} \; ; \; h_{c_i} - h_{e_{i,j}} \; ; \; h_{c_i} \cdot h_{e_{i,j}}\right] \quad (3)$$

where ";" denotes concatenation. The joint claim-evidence encodings are attention weighted and summed, projected through a fully connected layer into $\mathbb{R}^L$, where L is the number of possible labels:

$$o_i = \text{attn}([s_{i,1} \; ; \; ... \; ; \; s_{i,10}]) \quad (4)$$

$$p_i = \text{softmax}\left(\text{FC}(o_i)\right) \quad (5)$$

Lastly, the model is trained using cross entropy as the loss function. In the setting of using only the claim as the input (i.e., without the evidence), then h_{c_i} is used in Eq. 5 instead of o_i. If only the

[2] http://nlp.stanford.edu/data/glove.840B.300d.zip

RF (∼13 seconds)	Train: Snopes Within dataset Eval: Snopes $F1_{micro}$	 $F1_{macro}$	Out-of dataset Eval: PolitiFact $F1_{micro}$	 $F1_{macro}$	Train: PolitiFact Within dataset Eval: PolitiFact $F1_{micro}$	 $F1_{macro}$	Out-of dataset Eval: Snopes $F1_{micro}$	 $F1_{macro}$
Claim	0.473	0.231	**0.273**	0.223	0.254	0.255	0.546	0.243
Evidence	0.504	0.280	0.244	0.195	0.301	0.299	**0.597**	0.232
Claim+Evidence	0.550	0.271	0.245	0.190	0.310	0.304	0.579	0.207
LSTM (∼12 minutes, 888K parameters)								
Claim	0.408	0.243	0.260	**0.228**	0.237	0.237	0.565	0.221
Evidence	0.495	0.253	0.262	0.208	0.290	0.295	0.550	0.273
Claim+Evidence	0.529	0.253	0.258	0.189	0.288	0.294	0.509	0.256
BERT (∼264 minutes, 109M parameters)								
Claim	0.533	0.312	0.249	0.209	0.275	0.282	0.550	0.273
Evidence	0.531	**0.321**	0.249	0.224	**0.351**	**0.359**	0.577	**0.286**
Claim+Evidence	**0.556**	0.313	0.231	0.191	0.285	0.292	0.564	0.259

Table 1: Evaluation using micro and macro F1. Per column, the best score per method is underlined and the best score across all methods is highlighted in bold. We report the training time and number of model parameters, for Claim+Evidence on PolitiFact, in the parentheses. RF is trained on 5 cores and neural models on a Titan RTX.

evidence is used, then an attention weighted sum of the evidence snippet encodings is used in Eq. 5 instead of o_i.

BERT-based model. In a similar fashion to the LSTM model, we construct a model based on BERT (Devlin et al., 2019)[3], where the `[CLS]` token encoding is used for claim and evidence representations. Specifically, the claim and evidence snippets are encoded as:

$$h_{c_i} = \text{BERT}(c_i),\ h_{e_{i,j}} = \text{BERT}(c_i, e_{i,j}) \quad (6)$$

$$h_{e_i} = \text{attn}([h_{e_{i,1}}\ ;\ ...\ ;\ h_{e_{i,10}}]) \quad (7)$$

where the claim acts as the question when encoding the evidence snippets. Similarly to Eq. 5, the prediction is obtained by concatenating the claim and evidence representations and project it through a fully connected layer into $\mathbb{R}^L$:

$$p_i = \text{softmax}(FC([h_{c_i}\ ;\ h_{e_i}])) \quad (8)$$

where cross entropy is used as the loss function for training the model. In the setting that only the claim is used as input, then only h_{c_i} is used in Eq. 8. If only the evidence is used, then $h_{e_{i,j}}$ is computed without including c_i, and only h_{e_i} is used in Eq. 8.

[3]We use `bert-base-uncased` from https://huggingface.co/bert-base-uncased.

	#Claims	Labels
PolitiFact	13,581	pants on fire! (10.6%), false (19.2%), mostly false (17.0%), half-true (19.8%), mostly true (18.8%), true (14.8%)
Snopes	5,069	false (64.3%), mostly false (7.5%), mixture (12.3%), mostly true (2.8%), true (13.0%)

Table 2: Dataset statistics.

4 Experimental Evaluation

4.1 Datasets

We focus on the domain of political fact checking, where we use claims and associated evidence from PolitiFact and Snopes, which we extract from the MultiFC dataset (Augenstein et al., 2019). Using the claim as a query, the evidence is crawled from Google search API as the search snippets of the top-10 results, and is filtered such that the website origin of a given claim does not appear as evidence. To facilitate better comparison between the datasets, we filter claims with non-veracity related labels[4]. The dataset statistics are shown in Table 2.

4.2 Experimental setup

Both datasets are split into train/val/test sets using label-stratified sampling (70/10/20% splits). We tune all models on the validation split, and use early stopping with a patience of 10 for neural

[4]For PolitiFact we exclude [full flop, half flip, no flip] and for Snopes we exclude [unproven, miscaptioned, legend, outdated, misattributed, scam, correct attribution].

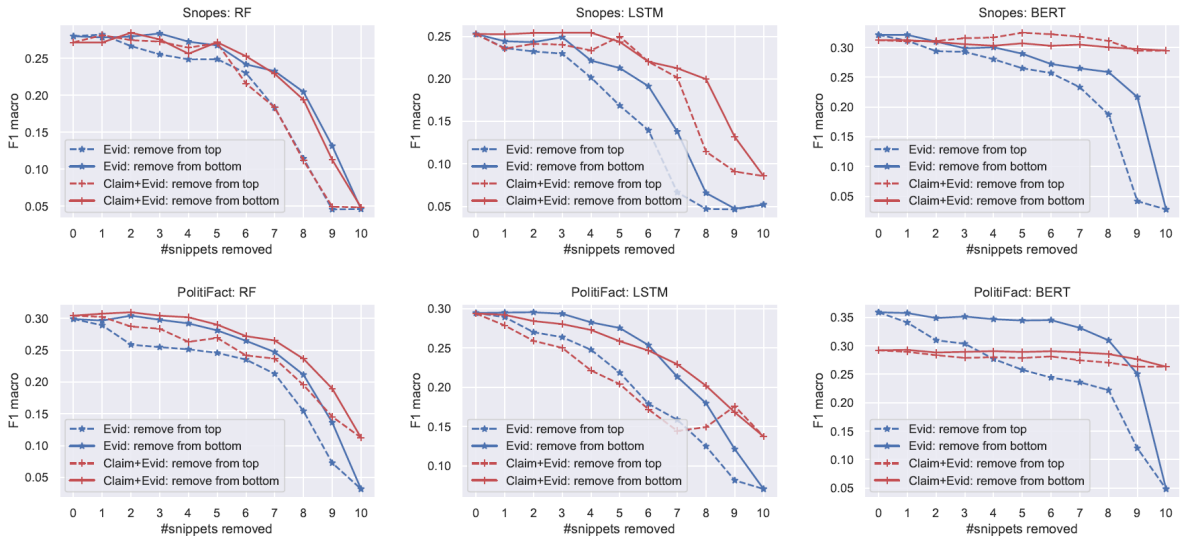


Figure 1: Macro F1 scores when removing evidence from either the top or bottom of the evidence snippet ranking.

models. Following Augenstein et al. (2019), we use micro and macro F1 for evaluation. The models are evaluated on both the within dataset test sets, but also out-of dataset test sets (e.g., a model trained on Snopes is evaluated on both Snopes and PolitiFact). In the out-of dataset evaluation we need the labels to be comparable, hence in that setting we merge "pants on fire!" and "false" for PolitiFact.

5 Tuning details

In the following, the best overall parameter configurations are underlined. The best configuration is chosen based on the average of the micro and macro F1[5]. For RF, we tune the number of trees from [100,500,1000], the minimum number of samples in a leaf from [1,3,5,10], and the minimum number of samples per split from [2,5,10]. For the LSTM model, we tune the learning rate from [1e-4,5e-4,1e-5], batch size [16,32], number of LSTM layers from [1,2], dropout from [0, 0.1], and fix the number of hidden dimensions to 128. For the BERT model, we tune the learning rate from [3e-5, 3e-6, 3e-7] and fix the batch size to 8.

5.1 Results

The results can be seen in Table 1. Overall, we see that the BERT model trained only on Evidence obtains the best results in 4/8 columns, and, notably, in 3/4 cases the BERT model with Evidence obtains the best macro F1 score on within and out-of dataset prediction. Random forest using term-frequency as input obtains the best out-of dataset micro F1 for both datasets (using either only Claim or only Evidence). Across all methods, the combination of Claim+Evidence only marginally obtains the best results a single time (for Snopes micro F1). For further details, in Table 3 we compute the accuracy scores for all the false labels, mixture or half-true label, and true labels.

Surprisingly, a BERT model using only the Evidence is capable of predicting the veracity of the claim used for obtaining the evidence. This shows that a strong signal must exist in the evidence itself, and the evidence found by the search engine appears to be implicitly affected by the veracity of the claim used as the query in some way[6]. The improvements reported in the literature by combining claim and evidence, are therefore not evident of the model learning to reason over the evidence with regards to the claim, but instead exploiting a signal inherent in the evidence itself. This highlights that the current approach for evidence gathering is problematic, as the strong signal makes it possible (and most often beneficial) for the model to entirely ignore the claim. This makes the model entirely reliant on the process behind how the evidence is generated, which is outside the scope of the model, and thereby undesirable, as any change in the search system may change the model performance significantly. It may also be problematic on a more fundamental level, e.g., to predict the

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[6] Note that the claim origin website is always removed from the evidence.

RF	Train: Snopes						Train: PolitiFact					
	Within dataset Eval: Snopes			Out-of dataset Eval: PolitiFact			Within dataset Eval: PolitiFact			Out-of dataset Eval: Snopes		
	acc_{false}	acc_{mix}	acc_{true}	acc_{false}	acc_{mix}	acc_{true}	acc_{false}	acc_{mix}	acc_{true}	acc_{false}	acc_{mix}	acc_{true}
Claim	0.710	0.144	0.255	0.853	0.016	0.209	0.623	0.216	0.513	0.790	0.092	0.255
Evidence	0.705	0.152	0.441	0.829	0.006	0.117	0.654	0.248	0.510	**0.891**	0.039	0.192
Claim+Evidence	**0.760**	0.136	0.453	0.829	0.000	0.117	0.634	0.292	0.512	0.871	0.039	0.199
LSTM												
Claim	0.674	0.232	0.280	0.875	0.047	0.137	0.566	0.212	0.504	0.833	0.026	0.234
Evidence	0.721	0.272	0.267	**0.890**	0.020	0.115	0.643	0.253	0.485	0.768	0.184	0.322
Claim+Evidence	0.757	0.248	0.168	0.879	0.008	0.107	**0.671**	0.210	0.460	0.704	0.171	**0.378**
BERT												
Claim	0.746	0.256	0.379	0.854	**0.094**	0.045	0.604	0.292	0.475	0.765	0.171	0.287
Evidence	0.648	**0.376**	**0.559**	0.702	0.049	**0.337**	0.649	**0.326**	0.496	0.804	**0.197**	0.339
Claim+Evidence	0.747	0.264	0.379	0.882	0.067	0.042	0.667	0.175	**0.558**	0.790	0.092	0.367

Table 3: Accuracy scores computed on the false labels, mixture or half-true label, and true labels. All labels within a group (e.g., any false label such as false or mostly false) are considered to be the same and as such this reduces the problem to a three class classification problem.

veracity of the following two claims: "the earth is round" and "the earth is flat", the evidence could be the same, but a model entirely dependent on the evidence, and not the claim, would be incapable of predicting both claims correctly.

5.2 Removal of evidence

We observed a strong predictive signal in the evidence alone and now consider the performance impact when gradually removing evidence snippets. The evidence is removed consecutively either from the top down or bottom up (i.e., removing the most relevant snippets first and vice versa), until no evidence is used. Figure 1 shows the macro F1 as a function of removed evidence when using the Evidence or Claim+Evidence models. We observe a distinct difference between the random forest and LSTM model compared to BERT: for random forest and LSTM, the Claim+Evidence models on both datasets drop rapidly in performance when the evidence is removed, while the BERT model only experiences a very small drop. This shows that when the Claim+Evidence is used in the BERT model, the influence of the evidence is minimal, while the evidence is vital for the Claim+Evidence RF and LSTM models. For all models, we observe that when evidence is removed from the top down, the performance drop is larger than when evidence is removed from the bottom up. Thus, the ranking of the evidence as provided by the search engine is related to its usefulness as evidence for fact checking.

6 Conclusion

We investigate if fact checking models for fake news detection are learning to process claim and evidence jointly in a way resembling reasoning. Across models of varying complexity and evaluated on multiple datasets, we find that the best performance can most often be obtained using only the evidence. This highlights that models using both claim and evidence are inherently not learning to reason, and points to a potential problem in how evidence is currently obtained in existing approaches for automatic fake news detection.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Lee Howell et al. 2013. Digital wildfires in a hyperconnected world. *WEF report*, 45(3):15–94.

Liesbeth Allein, Isabelle Augenstein, and Marie-Francine Moens. 2020. Time-aware evidence ranking for fact-checking. *arXiv preprint arXiv:2009.06402*.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4677–4691.

Ramy Baly, Mitra Mohtarami, James R. Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018. Integrating Stance Detection and Fact Checking in a Unified Corpus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 21–27.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019. Checkthat! at clef 2019: Automatic identification and verification of claims. In *Advances in Information Retrieval*, pages 309–315, Cham. Springer International Publishing.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Casper Hansen, Christian Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 994–1000.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, page 309–312, USA. Association for Computational Linguistics.

Nafise Sadat Moosavi and Michael Strube. 2017. Lexical features in coreference resolution: To be used with caution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Vancouver, Canada. Association for Computational Linguistics.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401. Association for Computational Linguistics.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32.

Aniketh Janardhan Reddy, Gil Rocha, and Diego Esteves. 2018. Defactonlp: Fact verification using entity recognition, TFIDF vector comparison and decomposable attention. *CoRR*, abs/1809.00509.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426. Association for Computational Linguistics.

Saying No is An Art: Contextualized Fallback Responses for Unanswerable Dialogue Queries

Ashish Shrivastava[1], Kaustubh D. Dhole[1], Abhinav Bhatt[2], Sharvani Raghunath[1]

[1]Amelia Science, IPsoft R&D [2]Universität des Saarlandes

[1]{firstname.lastname}@ipsoft.com

[2]abbh00001@stud.uni-saarland.de

Abstract

Despite end-to-end neural systems making significant progress in the last decade for task-oriented as well as chit-chat based dialogue systems, most dialogue systems rely on hybrid approaches which use a combination of rule-based, retrieval and generative approaches for generating a set of ranked responses. Such dialogue systems need to rely on a fallback mechanism to respond to out-of-domain or novel user queries which are not answerable within the scope of the dialogue system. While, dialogue systems today rely on static and unnatural responses like "I don't know the answer to that question" or "I'm not sure about that", we design a neural approach which generates responses which are contextually aware with the user query as well as say no to the user. Such customized responses provide paraphrasing ability and contextualization as well as improve the interaction with the user and reduce dialogue monotonicity. Our simple approach makes use of rules over dependency parses and a text-to-text transformer fine-tuned on synthetic data of question-response pairs generating highly relevant, grammatical as well as diverse questions. We perform automatic and manual evaluations to demonstrate the efficacy of the system.

1 Introduction

In order to cater to the diversity of questions spanning across various domains, dialogue systems generally follow a hybrid architecture wherein an ensemble of individual response subsystems (Kuratov et al.; Harrison et al., 2020) are employed from which an appropriate response is presented to the user (Serban et al., 2017; Finch et al., 2020; Paranjape et al., 2020). However, it is common for dialogue systems to encounter queries which are not within their scope of knowledge. While increasing the number of such subsystems would be a good strategy to increase coverage, it can be a never ending process and a default fallback strategy would always be needed. Besides, domain specific dialogue systems, especially those deployed in professional settings generally prefer restricting themselves to a fixed set of domains, and purposely refrain from responding to out-of-domain and random or toxic user queries.

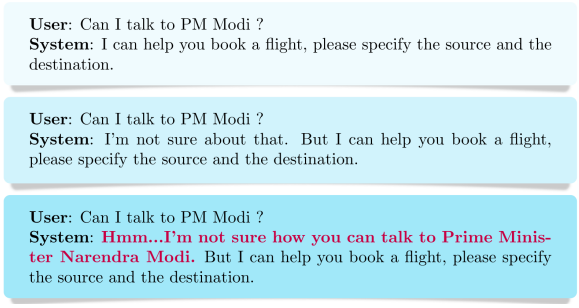


Figure 1: Comparison of responses of three flight booking dialogue systems: The first one does not handle unknown responses. The second one has a default fallback response. The third one has a fall-back response which is contextualized with the user query.

One approach to acknowledge such queries is to have a fallback mechanism with responses like "I don't know the answer to this question" or "I'm not sure how to answer that." However, such responses are static and unengaging and give an impression that the user's query has gone unacknowledged or is not understood by the system as shown in Figure 1 above.

Yu et al. (2016) have shown that static and predefined responses lead to lower levels of user engagement and decrease users' interest in interacting with the system. Yu et al. (2016) shows that a system which reacts to system breakdowns and to low user engagement leads to a better user engagement.

Our fallback approach attempts to address these limitations by generating "don't-know" responses which are engaging and contextually closer with the user query. 1) Since there are no publicly available datasets to generate such contextualised responses, we synthetically generate (query, fallback

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 87–92
August 1–6, 2021. ©2021 Association for Computational Linguistics

response) pairs using a set of highly accurate handcrafted dependency patterns. 2) We then train a sequence-to-sequence model over synthetic and natural paraphrases of these queries. 3) Finally, we measure the grammaticality and relevance of our models using a crowd-sourced setting to assess the generation capability. We have released the code and training dataset used in our experiments publicly. [1]

2 Related Work

Improving the coverage to address out-of-domain queries is not a new problem in designing dialogue systems. The most popular approach has been via presenting the user with chit-chat responses. Other systems such as Blender (Roller et al., 2020) and Meena (Adiwardana et al., 2020) promise to be successful for open-domain settings. Paranjape et al. (2020) finetune a GPT-2 model (Radford et al., 2019) on the EmpatheticDialogues dataset (Rashkin et al., 2019) to generate social talk responses. While this might seem fitting for chit-chat and social talk dialogue systems, domain-specific scenarios often dealing with professional settings would refrain from performing friendly or social talk especially avoiding the possibility of the randomness of generative models. Also, multiple subsystem architectures always have the possibility of cascading errors and profane or toxic queries. Hence systems should always have a foolproof mechanism in the form of static templates to reply from. Liang et al. (2020) uses an interesting approach for error handling by mapping dialogue acts and intents to templates. Besides, like Finch et al. (2020) it is always safer to generate fallback responses on encountering queries which might be toxic, biased or profane. [2]

Another line of work attempts to handle user queries which are ambiguous by asking back clarification questions (Dhole, 2020; Zamani et al., 2020; Yu et al., 2020). While this increases user interaction and coverage to an appreciable extent, it does not eliminate the requirement of a failsafe fallback responder. This paper's contribution is to address this requirement with an enhanced version of a fallback response generator.

[1] github.com/kaustubhdhole/natural-dont-know

[2] Handling programming exceptions and code failures also necessitates a simple fallback approach.

3 Methods

We describe two approaches to generate such contextual don't-know responses.

3.1 The Dependency Based Approach (DBA)

Inspired by previous approaches which use parse structures to generate questions (Heilman and Smith, 2009; Mazidi and Tarau, 2016; Dhole and Manning, 2020), we create a rule-based generator by handcrafting dependency templates to cater to a wide variety of question patterns as shown in Table 1. We perform extensive manual testing to improve the generations from these rules and increase overall coverage. The purpose of these rules is two-fold: i) To create a high-precision fall-back response generator as a baseline and ii) to help create (query, don't-know-response) pairs which could be paired with natural paraphrases to serve as seed training data for other deep learning architectures.

To build this baseline generator, we utilize few dependency templates in the style of SynQG (Dhole and Manning, 2020). We utilize the dependency parser from Andor et al. (2016) to get the Universal Dependencies (Nivre et al., 2016, 2017, 2020) of the user query. We then convert it to a don't-know-response by re-arranging nodes to a matched template. We further change pronouns, incorporate named entity information, and add rules to handle modals and auxiliaries. Finally, we also add rules for flipping pronouns to convert an agent targeted question to a user targeted response by interchanging pronouns and their supporting verbs. E.g. You to I and vice-versa.

We incorporate a bit of paraphrasing by randomizing various prefixes like "I'm not sure whether", "I don't know if", etc. and randomly using named entities. We describe the high-level algorithm below and in Algorithm 1.

$$prefix = pickRandom(prefixPool)$$
$$response = DBR(Question)$$
$$suffix = pickRandom(suffixPool)$$
$$fallbackResponse = Concat(prefix, response, suffix)$$

3.2 Sequence-to-Sequence Approach

Owing to the expected low coverage and scalability of the rule-based approach, we resort to take advantage of pre-trained neural architectures to attempt

Dependency Rules	Sample Question	Natural Don't Know Response
WhatBeRule()	What is the Pandora box ?	I am not really sure what the Pandora box is.
DidVerbRule()	Did Daniel cook today's meal ?	I don't know if Daniel cooked today's meal.
QuestionCanIRule()	Could you tell me the location of the tower ?	hmm..I don't know if I could tell you...
BeRule()	Are you predictive about conversation ?	I'm not sure if I am predictive about...
WhoBeRule()	Who is the Duke of Scotland ?	I can't be sure who the Duke of Scotland is.
WhoBeVerbRule()	Who is playing baseball and cricket both ?	I am not actually sure who is playing...
WhereBeRule()	Where did Bates translate this document ?	I don't know where Bates did translate...
HowBeRule()	How are the people of the Italy ?	I'm not sure how the people of that place are.
WhenBeRule()	When is the deadline of ACL ?	I can't be sure when the deadline of ACL is.
WhereBeVerbRule()	Where is Mr. Potter going ?	I'm not sure where Mr. Potter is going.
WhenBeVerbRule()	When will you submit your thesis ?	I'm not really sure when I will submit my thesis.

Table 1: Few Dependency Rules with the class of questions they cater too and their corresponding responses. In the 8^{th} sentence, the named entity "Italy" is randomly replaced by "that place".

Algorithm 1 Dependency Based Response (DBR)

```
nodes ← dependencyParse(Question)
for each template in templatePool do
    if (template condition matched) then
        Populate template using nodes
        Handle modals & auxilliaries
        Flip pronoun
        Randomly substitute Named Entity
if no template condition matched then
    return pickRandom(defaultResponse pool)
return filled template response
```

to create a sequence-to-sequence fallback responder. To incorporate noise and avoid the model to over-fit on the handcrafted transformations, we do not train the model directly on (query, don't-know-response) pairs generated from the previous section. From all possible questions of the Quora Questions Pairs dataset (QQP) [3], we first filter all the questions which generate a reply from the dependency based rules. Then we pair these dont-know-responses with the paraphrases of the input questions rather than the input questions themselves. [4] Primarily attempting to avoid over-fitting on the dependency patterns, this also helps generate dont-know-responses which are paraphrastic in nature.

After incorporating paraphrases from QQP, we are able to build a dataset of 100k pairs, which we call the "I Dont Know Dataset" (IDKD). After witnessing the success of text-to-text transformers, we use the pre-trained T5 transformer (Raffel et al., 2020a,b) as our sequence-to-sequence model. We divide IDKD into a train and validation split of 80:20. We use the Transformers code from HuggingFace (Wolf et al., 2020) to fine-tune a T5-base model over IDKD for 2 epochs. [5]

Metrics	DBA	Seq-To-Seq
%GC	81.6	87.2
ARS	3.97	3.66

Table 2: Human evaluation between the two approaches. %GC= % of Grammatically correct responses, ARS=Average Relevance Score.

4 Results

Most prior generated systems are evaluated on a range of automatic metrics like BLEU and ROGUE (Papineni et al., 2002) used in the machine translation literature. However, owing to the drawbacks of these metrics, we perform human evaluation of the generated responses using two metrics - namely "relevance" and "grammaticality" as defined in Dhole and Manning (2020). We evaluate the performance of both the approaches in a crowd-sourced setting by requesting English-schooled individuals to rate. [6] Raters were asked to evaluate grammaticality in a binary setting (grammatical/ungrammatical) and relevance on a Likert scale (1 to 5).

Our human evaluations are shown in Table-2. T5 responses tend to be more grammatical than their dependency counterparts by a large margin of 6%. Relevance scores drop slightly from 3.97 to 3.66.

[3] Quora Question Pairs Dataset

[4] Those question pairs which have the label "1" or are similar are used as paraphrases.

[5] Increasing the epochs to 3 and 4 tended to overfit and not generate paraphrases.

[6] Our pool of English-schooled raters consisted of engineering graduates having attained high school and engineering education with English as a medium of instruction.

Metrics	Question	Dependency Based Approach	Seq2Seq Approach
Coverage	–	54.56%	100%
Average Sentence Length	9.41	11.83	11.78
Sentence Length Variance	11.74	11.60	10.66
Average #Novel Words	–	0.91	2.57

Table 3: System evaluation between Dependency Based Approach and Sequence to Sequence Approach.

Question: Are you excited about natural language processing research ?
DBA: I'm not sure if I am excited about natural language processing research.
Seq2Seq: I'm not actually sure what the benefits of natural language processing are.

Question: What is the quickest way to increase Instagram followers?
DBA: I am not sure what the quickest way to increase Instagram followers is.
Seq2Seq: I cannot be sure how a user can increase Instagram followers.

Question: How can I join MIT?
DBA: I am not sure how you can join MIT.
Seq2Seq: I am not sure what the eligibility criteria for MIT are.

Question: How can I drive more traffic to my website?
DBA: I'm not really sure how you can drive more traffic.
Seq2Seq: I am not sure what the best way to get free traffic to your website is.

Question: Where can I find a collection of poetry by William Shakespeare?
DBA: I'm not really sure where you can find a collection of poetry by William Shakespeare.
Seq2Seq: I'm not actually sure where you can get a collection of his poems.

Question: What is the effect of demonetization of 500 and 1000 rupees notes?
DBA: I cannot be sure what the effect of demonetization of 500 and 1000 rupees notes is.
Seq2Seq: I'm not really sure what my views on PM Modi's decision on discontinuing 500 & 1000 rs notes are.

Table 4: Natural don't know responses generated by both the approaches. Highlights in blue depict words, phrases or events not mentioned by the user.

This can be largely attributed to the model's paraphrastic ability of describing words and connected events outside the knowledge of the user's query. Eg. in the second query in Table 4, if the string "MIT" were something other than an institution, the dependency based approach would seem safer than the seq2seq approach.

In addition, T5 responses on an average generate at least double the number of novel words than their dependency counterparts as shown in Table 3. Sentence length mostly remains unaffected across the two models. Undoubtedly, the rule-based model despite being highly relevant is only able to reply to 54.5% of random QQP queries.

The T5 model helped to not only add paraphrastic variations but also scale to user queries outside of the scope of the dependency templates. More importantly, without losing the original ability of saying no, the model was able to generate more natural sounding dont-know-reponses by utilizing it's inherent world-knowledge acquired during pre-training. Table 4 shows some interesting examples. The highlighted phrases in blue show the benefits of the model's pre-training ability.

5 Conclusion and Future work

We describe two simple approaches which enhance user interaction to cater to the necessities of real-life dialogue systems which are generally a tapestry of multiple solitary subsystems. In order to avoid cascading errors from such systems, as well as refrain from answering out-of-domain and toxic queries it is but natural to have a fallback approach to say no. We argue that such a fallback approach could be contextualised to generate engaging responses by having multiple ways of saying no rather than a one common string for all approach. The appeal of our approach is the ease with which

it can rightly fit within any larger dialogue design framework.

Of course, this is not to deny that as we give more paraphrasing power to the fallback system, it would tend to retract from succinctly replying with a no - as is evident from the drop in the relevance scores. Nevertheless, we still believe that both our fallback approaches could serve as effective baselines for future work.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.

Kaustubh Dhole and Christopher D. Manning. 2020. Syn-QG: Syntactic and shallow semantic rules for question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 752–765, Online. Association for Computational Linguistics.

Kaustubh D Dhole. 2020. Resolving Intent Ambiguities by Retrieving Discriminative Clarifying Questions. *arXiv preprint arXiv:2008.07559*.

Sarah E Finch, James D Finch, Ali Ahmadvand, Xiangjue Dong, Ruixiang Qi, Harshita Sahijwani, Sergey Volokhin, Zihan Wang, Zihao Wang, Jinho D Choi, et al. 2020. Emora: An inquisitive social chatbot who cares for you. *arXiv preprint arXiv:2009.04617*.

Vrindavan Harrison, Juraj Juraska, Wen Cui, Lena Reed, Kevin K Bowden, Jiaqi Wu, Brian Schwarzmann, Abteen Ebrahimi, Rishi Rajasekaran, Nikhil Varghese, et al. 2020. Athena: Constructing dialogues dynamically with discourse constraints. *arXiv preprint arXiv:2011.10683*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon Univ Pittsburgh pa language technologies insT.

Yuri Kuratov, Idris Yusupov, Dilyara Baymurzina, Denis Kuznetsov, Daniil Cherniavskii, Alexander Dmitrievskiy, Elena Ermakova, Fedor Ignatov, Dmitry Karpov, and Daniel Kornev. Dream technical report for the alexa prize 2019.

Kaihui Liang, Austin Chau, Yu Li, Xueyuan Lu, Dian Yu, Mingyang Zhou, Ishan Jain, Sam Davidson, Josh Arnold, Minh Nguyen, et al. 2020. Gunrock 2.0: A user adaptive social conversational system. *arXiv preprint arXiv:2011.08906*.

Karen Mazidi and Paul Tarau. 2016. Infusing NLU into automatic question generation. In *Proceedings of the 9th International Natural Language Generation conference*, pages 51–60, Edinburgh, UK. Association for Computational Linguistics.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, et al. 2017. Universal dependencies 2.1.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. 2020. Recipes for building an open-domain chatbot.

Iulian V. Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, Sai Rajeshwar, Alexandre de Brebisson, Jose M. R. Sotelo, Dendi Suhubdy, Vincent Michalski, Alexandre Nguyen, Joelle Pineau, and Yoshua Bengio. 2017. A deep reinforcement learning chatbot.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lili Yu, Howard Chen, Sida I. Wang, Tao Lei, and Yoav Artzi. 2020. Interactive classification by asking informative questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2664–2680, Online. Association for Computational Linguistics.

Zhou Yu, Leah Nicolich-Henkin, Alan W Black, and Alexander Rudnicky. 2016. A Wizard-of-Oz study on a non-task-oriented dialog systems that reacts to user engagement. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 55–63, Los Angeles. Association for Computational Linguistics.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, pages 418–428.

N-Best ASR Transformer: Enhancing SLU Performance using Multiple ASR Hypotheses

Karthik Ganesan*, **Pakhi Bamdev***, **Jaivarsan B***, **Amresh Venugopal, Abhinav Tushar**
Vernacular.ai
{karthikganesan17, pakhi.bamdev.in}@gmail.com
{jaivarsan, amresh, abhinav}@vernacular.ai

Abstract

Spoken Language Understanding (SLU) systems parse speech into semantic structures like dialog acts and slots. This involves the use of an Automatic Speech Recognizer (ASR) to transcribe speech into multiple text alternatives (hypotheses). Transcription errors, common in ASRs, impact downstream SLU performance negatively. Approaches to mitigate such errors involve using richer information from the ASR, either in form of N-best hypotheses or word-lattices. We hypothesize that transformer models learn better with a simpler utterance representation using the concatenation of the N-best ASR alternatives, where each alternative is separated by a special delimiter [SEP]. In our work, we test our hypothesis by using concatenated N-best ASR alternatives as the input to transformer encoder models, namely BERT and XLM-RoBERTa, and achieve performance equivalent to the prior state-of-the-art model on DSTC2 dataset. We also show that our approach significantly outperforms the prior state-of-the-art when subjected to the low data regime. Additionally, this methodology is accessible to users of third-party ASR APIs which do not provide word-lattice information.

1 Introduction

Spoken Language Understanding (SLU) systems are an integral part of Spoken Dialog Systems. They parse spoken utterances into corresponding semantic structures e.g. dialog acts. For this, a spoken utterance is usually first transcribed into text via an Automated Speech Recognition (ASR) module. Often these ASR transcriptions are noisy and erroneous. This can heavily impact the performance of downstream tasks performed by the SLU systems.

To counter the effects of ASR errors, SLU systems can utilise additional feature inputs from ASR. A common approach is to use N-best hypotheses where multiple ranked ASR hypotheses are used, instead of only 1 ASR hypothesis. A few ASR systems also provide additional information like word-lattices and word confusion networks. Word-lattice information represents alternative word-sequences that are likely for a particular utterance, while word confusion networks are an alternative topology for representing a lattice where the lattice has been transformed into a linear graph. Additionally, dialog context can help in resolving ambiguities in parses and reducing impact of ASR noise.

N-best hypotheses: Li et al. (2019) work with 1-best ASR hypothesis and exploits unsupervised ASR error adaption method to map ASR hypotheses and transcripts to a similar feature space. On the other hand, Khan et al. (2015) uses multiple ASR hypotheses to predict multiple semantic frames per ASR choice and determine the true spoken dialog system's output using additional context. **Word-lattices:** Ladhak et al. (2016) propose using recurrent neural networks (RNNs) to process weighted lattices as input to SLU. Švec et al. (2015) presents a method for converting word-based ASR lattices into word-semantic (W-SE) which reduces the sparsity of the training data. Huang and Chen (2019) provides an approach for adapting lattices with pretrained transformers. **Word confusion networks (WCN):** Jagfeld and Vu (2017) proposes a technique to exploit word confusion networks (WCNs) as training or testing units for slot filling. Masumura et al. (2018) models WCN as sequence of bag-of-weighted-arcs and introduce a mechanism that converts the bag-of-weighted-arcs into a continuous representation to build a neural network based spoken utterance classification. Liu et al. (2020) proposes a BERT based SLU model to encode WCNs and the dialog context jointly to

* The first three authors have equal contribution.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 93–98
August 1–6, 2021. ©2021 Association for Computational Linguistics

reduce ambiguity from ASR errors and improve SLU performance with pre-trained models.

The motivation of this paper is to improve performance on downstream SLU tasks by exploiting *transfer learning* capabilities of the pre-trained transformer models. Richer information representations like word-lattices (Huang and Chen (2019)) and word confusion networks (Liu et al. (2020)) have been used with GPT and BERT respectively. These representations are non-native to Transformer models, that are pre-trained on plain text sequences. We hypothesize that transformer models will learn better with a simpler utterance representation using concatenation of the N-best ASR hypotheses, where each hypothesis is separated by a special delimiter [SEP]. We test the effectiveness of our approach on a dialog state tracking dataset - DSTC2 (Henderson et al., 2014), which is a standard benchmark for SLU.

Contributions: (i) Our proposed approach, trained with a simple input representation, exceeds the competitive baselines in terms of accuracy and shows equivalent performance on the F1-score to the prior state-of-the-art model. (ii) We significantly outperform the prior state-of-the-art model in the low data regime. We attribute this to the effective *transfer learning* from the pre-trained Transformer model. (iii) This approach is accessible to users of third party ASR APIs unlike the methods that use word-lattices and word confusion networks which need deeper access to the ASR system.

2 N-Best ASR Transformer

N-Best ASR Transformer[1] works with a simple input representation achieved by concatenating the N-Best ASR hypotheses together with the dialog context (system utterance). Pre-trained transformer models, specifically BERT and XLMRoBERTa, are used to encode the input representation. For output layer, we use a semantic tuple classifier (STC) to predict *act-slot-value* triplets. The following sub-sections describe our approach in detail.

2.1 Input Representation

For representing the input we concatenate the last system utterance S (dialog context), and the user utterance U. U is represented as concatenation of the N-best[2] ASR hypotheses, separated by a special delimiter, [SEP]. The final representation is shown in equation 1 below:

$$x_i = [\text{CLS}] \oplus \text{TOK}(S_i) \oplus \bigoplus_{j=1}^{N} (\text{TOK}(U_i^j) \oplus [\text{SEP}]) \quad (1)$$

Here, U_i^j refers to the j^{th} ASR hypothesis for the i^{th} sample, $\oplus$ denotes the concatenation operator, TOK(.) is the tokenizer, [CLS] and [SEP] are the special tokens.

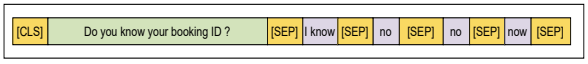


Figure 1: Input representation: The green boxes represents the last system utterances followed by ASR hypotheses of user utterances concatenated together with a [SEP] token.

As represented in figure 2, we also pass segment IDs along with the input to differentiate between segment a (last system utterance) and segment b (user utterance).

2.2 Transformer Encoder

The above mentioned input representation can be easily used with any pre-trained transformer model. For our experiments, we select BERT (Devlin et al., 2019) and XLM-RoBERTa[3] (Conneau et al., 2020) for their recent popularity in NLP research community.

2.3 Output Representation

The final hidden state of the transformer encoder corresponding to the special classification token [CLS] is used as an aggregated input representation for the downstream classification task by a semantic tuple classifier (STC) (Mairesse et al., 2009). STC uses two classifiers to predict the *act-slot-value* for a user utterance. A binary classifier is used to predict the presence of each *act-slot* pair, and a multi-class classifier is used to predict the *value* corresponding to the predicted act-slot pairs. We omit the latter classifier for the act-slot pairs with no value (like *goodbye*, *thankyou*, *request_food* etc.).

[1]The code is available at https://github.com/Vernacular-ai/N-Best-ASR-Transformer

[2]We use ASR transcriptions (N $\leq$ 10) provided by DSTC2 dataset to perform our experiments. Our input structure can support variable N during training and inference.

[3]The model name XLM-RoBERTa and XLM-R will be used interchangeably throughout the paper.

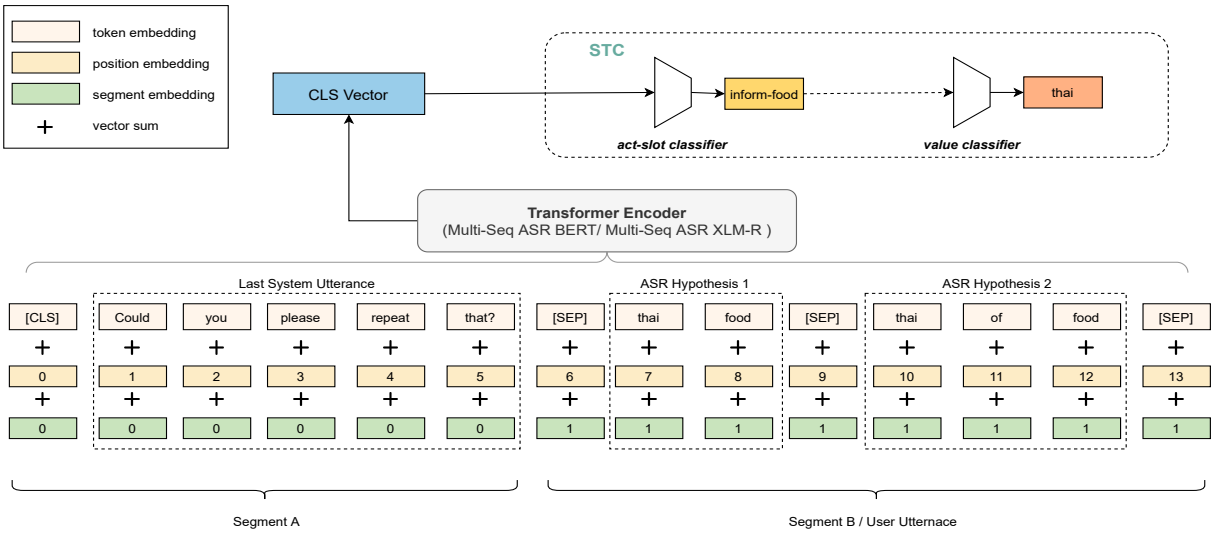


Figure 2: **N-Best ASR Transformer:** The input representation is encoded by a transformer model which forms an input for a Semantic Tuple Classifier (STC). STC uses binary classifiers to predict the presence of act-slot pairs, followed by a multi-class classifier that predicts the value for each act-slot pair.

3 Experimental Setup

3.1 Dataset

We perform our experiments on data released by the Dialog State Tracking Challenge (DSTC2) (Henderson et al., 2014). It includes pairs of utterances and the corresponding set of *act-slot-value* triplets for training (11,677 samples), development (3,934 samples), and testing (9,890 samples). The task in the dataset is to parse the user utterances like "*I want a moderately priced restaurant.*" into a corresponding semantic representation in the form of "*inform(pricerange=moderate)*" triplet. For each utterance, both the manual transcription and a maximum of 10-best ASR hypotheses are provided. The utterances are annotated with multiple *act-slot-value* triplets. For transcribing the utterances DSTC2 uses two ASRs - one with an artificially degraded statistical acoustic model, and one which is fully optimized for the domain. Training and development sets include transcriptions from both the ASRs. To utilise this dataset we first transform it into the input format as discussed in section 2.1.

3.2 Baselines

We compare our approach with the following baselines:

- **SLU2** (Williams, 2014): Two binary classifiers (decision trees) are used with word n-grams from the ASR N-best list and the word confusion network. One predicts the presence of that slot-value pair in the utterance and the other estimate for each user dialog act.

- **CNN+LSTMw4** (Rojas-Barahona et al., 2016): A convolution neural network (CNN) is trained with the N-best ASR hypotheses to output the utterance representation. A long-short term memory network (LSTM) with a context window size of 4 outputs a context representation. The models are jointly trained to predict for the act-slot pair. Another model with the same architecture is trained to predict for the value corresponding to the predicted act-slot pair.

- **CNN** (Zhao and Feng, 2018): Proposes CNN based models for dialog act and slot-type prediction using 1-best ASR hypothesis.

- **Hierarchical Decoding** (Zhao et al., 2019): A neural-network based binary classifier is used to predict the act and slot type. A hybrid of sequence-to-sequence model with attention and pointer network is used to predict the value corresponding to the detected act-slot pair.1-Best ASR hypothesis was used for both training and evaluation tasks.

- **WCN-BERT + STC** (Liu et al., 2020): Input utterance is encoded using the Word Confusion Network (WCN) using BERT by having the same position ids for all words in the bin of a lattice and modifying self-attention to work with word probabilities. A semantic tuple classifier uses a binary classifier to predict the act-slot value, followed by a multi-class classifier that predicts the value corresponding

to the act-slot tuple.

3.3 Experimental Settings

We perform hyper-parameter tuning on the validation set to get optimal values for dropout rate δ, learning rate lr, and the batch size b. Based on the best F1-score, the final selected parameters were δ = 0.3, lr = 3e-5 and b = 16. We set the warm-up rate wr = 0.1, and L2 weight decay $L2$ = 0.01. We make use of Huggingface's *Transformers* library (Wolf et al., 2020) to fine-tune the *bert-base-uncased* and *xlm-roberta-base*, which is optimized over Huggingface's BertAdam optimizer. We trained the model on Nvidia T4 single GPU on AWS EC2 g4dn.2xlarge instance for 50 epochs. We apply early stopping and save the best-performing model based on its performance on the validation set.

4 Results

In this section, we compare the performance of our approach with the baselines on the DSTC2 dataset. To compare the *transfer learning* effectiveness of pre-trained transformers with *N-Best ASR BERT* (our approach) and the previous state-of-the-art model *WCN-BERT STC*, we perform comparative analysis in the low data regime. Additionally, we perform an ablation study on *N-Best ASR BERT* to see the impact of modeling dialog context (last system utterance) with the user utterances.

4.1 Performance Evaluation

Model	F1-score	Accuracy
SLU2	82.1	-
CNN+LSTM_w4	83.6	-
CNN	85.3	-
Hierarchical Decoding	86.9	-
WCN-BERT + STC	**87.9**	81.1
N-Best ASR XLM-R (Ours)	87.4	**81.9**
N-Best ASR BERT (Ours)	87.8	81.8

Table 1: F1-scores (%) and utterance-level accuracy (%) of baseline models and our proposed model on the test set.

Since the task is a multi-label classification of *act-slot-value* triplets, we report utterance level accuracy and F1-score. A prediction is correct if the set of labels predicted for a sample exactly matches the corresponding set of labels in the ground truth. As shown in Table 1, we compare our models, *N-Best ASR BERT* and *N-Best ASR XLM-R*, with baselines mentioned in section . Both of our proposed models, trained with concatenated N-Best ASR hypotheses, outperform the competitive baselines in terms of accuracy and show comparable performance on F1-score with *WCN-BERT STC*.

4.2 Performance in Low Data Regime

Train Data (%age)	WCN-BERT STC	*N-Best ASR BERT*
5	78.5	**83.9**
10	80.3	**85.5**
20	84.4	**86.7**
50	85.9	**87.7**

Table 2: F1-scores (%) for our proposed model *N-Best ASR BERT* (ours) and *WCN-BERT STC* (previous state-of-the-art.

To study the performance of model in the low data regime, we randomly select p percentage of samples from the training set in a stratified fashion, where $p \in \{5, 10, 20, 50\}$. We pick our model *N-Best ASR BERT* and *WCN-BERT STC* for this study because both use BERT as the encoder model. For both models, we perform experiments using the same training, development, and testing splits. From Table 2, we find that *N-Best ASR BERT* outperforms *WCN-BERT STC* model significantly for low data regime, especially when trained on 5% and 10% of the training data. It shows that our approach effectively *transfer learns* from pre-trained transformer's knowledge. We believe this is due to the structural similarity between our input representation and the input BERT was pre-trained on.

4.3 Significance of Dialog Context

Model	Variation	F1-score	Accuracy
N-Best ASR BERT	without system utterance	86.5	80.2
	with system utterance	**87.8**	**81.8**

Table 3: F1-scores (%) and utterance-level accuracy (%) of our model *N-Best ASR BERT* on the test set when trained with and without system utterances.

Through this ablation study, we try to understand the impact of dialog context on model's performance. For this, we train *N-Best ASR BERT* in the following two settings:

- When input representation consists of only the user utterance.
- When input representation consists of both the last system utterance (dialog context) and the user utterance as shown in figure 3.

As presented in Table 3, we observe that modeling the last system utterance helps in achieving better F1 and utterance-level accuracy by the difference of 1.3% and 1.6% respectively.

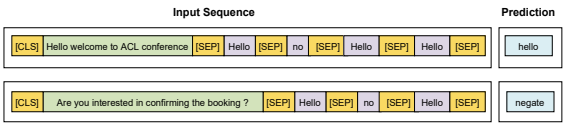


Figure 3: Significance of Dialog Context: The green box depicts the dialog context that helps disambiguate the very similar ASR hypotheses shown in purple boxes.

It proves that dialog context helps in improving the performance of downstream SLU tasks. Figure 3 represents one such example where having dialog context in form of the last system utterance helps disambiguate between the two similar user utterances.

5 Conclusion

In this work, building on a simple input representation, we propose *N-Best ASR Transformer*, which outperforms all the competitive baselines on utterance-level accuracy for the DSTC2 dataset. However, the highlight of our work is in achieving significantly higher performance in an extremely low data regime. This approach is accessible to users of third-party ASR APIs, unlike the methods that use word-lattices and word confusion networks. As future extensions to this work, we plan to :

- Enable our proposed model to generalize to out-of-vocabulary (OOV) slot values.
- Evaluate our approach in a multi-lingual setting.
- Evaluate on different values N in N-best ASR.
- Compare the performance of our approach on ASRs with different Word Error Rates (WERs).

Acknowledgement

We are highly grateful to our organization Vernacular.ai and our Machine Learning Team for (i) exposing us to practical problems related to multilingual voice-bots, (ii) giving us access to resources to solve this problem, (iii) helping us deploy this work in production for real-world users, and (iv) for their excellent feedback on this work.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

Chao-Wei Huang and Yun-Nung Chen. 2019. Adapting pretrained transformer to lattices for spoken language understanding. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 845–852. IEEE.

Glorianna Jagfeld and Ngoc Thang Vu. 2017. Encoding word confusion networks with recurrent neural networks for dialog state tracking. In *Proceedings of the Workshop on Speech-Centric Natural Language Processing*, pages 10–17, Copenhagen, Denmark. Association for Computational Linguistics.

Omar Zia Khan, Jean-Philippe Robichaud, Paul A Crook, and Ruhi Sarikaya. 2015. Hypotheses ranking and state tracking for a multi-domain dialog system using multiple asr alternates. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Faisal Ladhak, Ankur Gandhe, Markus Dreyer, Lambert Mathias, Ariya Rastrow, and Björn Hoffmeister. 2016. Latticernn: Recurrent neural networks over lattices. In *Interspeech*, pages 695–699.

Hao Li, Chen Liu, Su Zhu, and Kai Yu. 2019. Robust spoken language understanding with acoustic and domain knowledge. In *2019 International Conference on Multimodal Interaction*, pages 531–535.

Chen Liu, Su Zhu, Zijian Zhao, Ruisheng Cao, Lu Chen, and Kai Yu. 2020. Jointly encoding word confusion network and dialogue context with BERT for spoken language understanding. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 871–875. ISCA.

François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4749–4752. IEEE.

Ryo Masumura, Yusuke Ijima, Taichi Asami, Hirokazu Masataki, and Ryuichiro Higashinaka. 2018. Neural confnet classification: Fully neural network based spoken utterance classification using word confusion networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6039–6043. IEEE.

Lina M. Rojas-Barahona, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, Stefan Ultes, Tsung-Hsien Wen, and Steve Young. 2016. Exploiting sentence and context representations in deep neural models for spoken language understanding. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 258–267, Osaka, Japan. The COLING 2016 Organizing Committee.

Jan Švec, Luboš Šmídl, Tomáš Valenta, Adam Chỳlek, and Pavel Ircing. 2015. Word-semantic lattices for spoken language understanding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5266–5270. IEEE.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 282–291.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lin Zhao and Zhe Feng. 2018. Improving slot filling in spoken language understanding with joint pointer and attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 426–431.

Zijian Zhao, Su Zhu, and Kai Yu. 2019. A hierarchical decoding model for spoken language understanding from unaligned data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7305–7309. IEEE.

Gender Bias Amplification During Speed-Quality Optimization in Neural Machine Translation

Adithya Renduchintala, Denise Diaz[*1], Kenneth Heafield, Xian Li, Mona Diab
Facebook AI, [1]Independent Researcher
{adirendu,kheafield,xianl,mdiab}@fb.com
denisedeediaz@gmail.com

Abstract

Is bias amplified when neural machine translation (NMT) models are optimized for speed and evaluated on generic test sets using BLEU? We investigate architectures and techniques commonly used to speed up decoding in Transformer-based models, such as greedy search, quantization, average attention networks (AANs) and shallow decoder models and show their effect on gendered noun translation. We construct a new gender bias test set, SimpleGEN, based on gendered noun phrases in which there is a single, unambiguous, correct answer. While we find minimal overall BLEU degradation as we apply speed optimizations, we observe that gendered noun translation performance degrades at a much faster rate.

1 Introduction

Optimizing machine translation models for production, where it has the most impact on society at large, will invariably include speed-accuracy trade-offs, where accuracy is typically approximated by BLEU scores (Papineni et al., 2002) on generic test sets. However, BLEU is notably not sensitive to specific biases such as gender. Even when speed optimizations are evaluated in shared tasks, they typically use BLEU (Papineni et al., 2002; Heafield et al., 2020) to approximate quality, thereby missing gender bias. Furthermore, these biases probably evade detection in shared tasks that focus on quality without a speed incentive (Guillou et al., 2016) because participants would not typically optimize their systems for speed. Hence, it is not clear if Neural Machine Translation (NMT) speed-accuracy optimizations amplify biases. This work attempts to shed light on the *algorithmic choices* made during speed-accuracy optimizations and their impact on gender biases in an NMT system, complementing existing work on data bias.

source	That physician is a funny lady!
reference	¡Esa médica/doctora es una mujer graciosa!
system A	¡Ese médico es una dama graciosa!
system B	¡Ese médico es una dama divertida!
system C	¡Ese médico es una mujer divertida!
system D	¡Ese médico es una dama divertida!

Table 1: Translation of a simple source sentence by 4 different commercial English to Spanish MT systems. All of these systems fail to consider the token "lady" when translating the occupation-noun, rendering it in with the masculine gender "doctor/médico".

We explore optimizations choices such as (i) search (changing the beam size in beam search); (ii) architecture configurations (changing the number of encoder and decoder layers); (iii) model based speedups (using Averaged attention networks (Zhang et al., 2018)); and (iv) 8-bit quantization of a trained model..

Prominent prior work on gender bias evaluation forces the system to "guess" the gender (Stanovsky et al., 2019a) of certain occupation nouns in the source sentence. Consider, the English source sentence "That physician is funny.", containing no information regarding the physician's gender. When translating this sentence into Spanish (where the occupation nouns are explicitly specified for gender), an NMT model is forced to guess the gender of the physician and choose between masculine forms, doctor/médico or feminine forms doctora/médica. While investigating bias in these settings is valuable, in this paper, we hope to highlight that the problem is much worse — despite an explicit gender reference in the sentence, NMT systems still generate the wrong gender in translation (see Table 1), resulting in egregious errors where not only is the gender specification incorrect but the generated sentence also fails in morphological gender

[*]This work conducted while author was working at Facebook AI.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 99–109
August 1–6, 2021. ©2021 Association for Computational Linguistics

Templates			That `f/m-occ-sg` is a funny `f/m-n-sg`! My `f/m-rel` is a `f/m-occ-sg`.
Keywords			`f-occ-sg` = {nurse, nanny...} `m-occ-sg` = {physician, mechanic...} `f-rel` = {sister, mother..} `m-rel` = {brother, father...} `f-n-sg` = {woman, gal, lady...} `m-n-sg` = {man, guy...}
Generated	pro.	MoMc	That engineer is a funny guy! My father is a mechanic.
		FoFc	That nanny is a funny lady! My mother is a nurse.
	anti.	MoFc	That mechanic is my funny woman! My sister is a physician.
		FoMc	That nurse is funny man! My brother is a nanny.

Table 2: Example Templates, Keywords and a sample of the resulting generated source sentences.

agreement. To focus on these egregious errors, we construct a new data set, SimpleGEN. In SimpleGEN, all source sentences include an occupation noun (such as "mechanic", "nurse" etc.) *and* an unambiguous "signal" specifying the gender of the person being referred to by the occupation noun. For example, we modify the previous example to "That physician is a funny *lady*". We call our dataset "Simple" because it contains all the information needed by a model to produce correctly gendered occupation nouns. Furthermore, our sentences are short (up to 12 tokens) and do not contain complicated syntactic structures. Ideally, SimpleGEN should obviate the need for an NMT model to incorrectly guess the gender of occupation nouns, but using this dataset we show that gender translation accuracy, particularly in female context sentences (see Section 2), is negatively impacted by various speed optimizations at a *greater rate* than a drop in BLEU scores. A small drop in BLEU can hide a large increase in biased behavior in an NMT system. Further illustrating how insensitive BLEU is as a metric to such biases.

2 SimpleGEN: A gender bias test set

Similar to Stanovsky et al. (2019b), our goal is to provide English input to an NMT model and evaluate if it correctly genders occupation-nouns. We focus on English to Spanish (En-Es) and English to German (En-De) translation directions as occupation-nouns are explicitly specified for gender in these target languages while English is underspecified for such a morphological phenomenon which forces the model to attend to contextual clues. Furthermore, these language directions are considered "high-resource" and often cited as exemplars for advancement in NMT.

A key differentiating characterization of our test set is that there is no ambiguity about the gender of the occupation-noun. We achieve this by using carefully constructed templates such that there is enough contextual evidence to *unambiguously specify* the gender of the occupation-noun. Our templates specify a "scaffolding" for sentences with *keywords* acting as placeholders for *values* (see Table 2). For the occupation keywords such as `f-occ-sg` and `m-occ-sg`, we select the occupations for our test set using the U.S Department of Labor statistics of high-demand occupations.[1] A full list of templates, keywords and values is in table A6. Using our templates, we generate English source sentences which fall into two categories: (i) *pro-stereotypical* (pro) sentences contain either stereotypical male occupations situated in male contexts (MOMC) or female occupations in female contexts (FOFC), and (ii) *anti-stereotypical* (anti) sentences in which the context gender and occupation gender are mismatched, i.e. male occupations in female context (MOFC) and female occupations in male contexts (FOMC). Note that we use the terms "male context" or "female context" to categorize sentences in which there is an unambiguous signal that the occupation noun refers to a male or female person, respectively. We generated 1332 pro-stereotypical and anti-stereotypical sentences, 814 in the MOMC and MOFC subgroups and 518 in the FOMC and FOFC subgroups (we collect more male stereotypical occupations compared to female, which causes this disparity).

To evaluate the translations of NMT models on SimpleGEN, we also create an occupation-noun bilingual dictionary, that considers the number and gender as well as synonyms for the occupations. For example for the En-Es direction, the English occupation term 'physician", has corresponding entries for its feminine forms in Spanish as "doctora" and "médica" and for its masculine forms "doctor" and "médico" (See table A8 for our full dictionary). By design, non-occupation keywords such as `f-rel` and `f-n-sg` specify the expected gender of the occupation-noun on the target side, enabling dictionary based correctness verification.

3 Speeding up NMT

There are several "knobs" that can be tweaked to speed up inference for NMT models. Setting the beam-size (bs) to 1 during beam search is likely the

[1] https://www.dol.gov/agencies/wb/data/high-demand-occupations

Source	That physician is a funny lady!	Label
Translations	¡Esa doctora es una mujer graciosa!	Correct
	¡Esa médica es una mujer feliz!	Correct
	¡Ese médico es una mujer graciosa!	Incorrect
	¡Ese medicación es una mujer graciosa!	NA

Table 3: Our evaluation protocol with an example source sentence and four example translations.

simplest approach to obtain quick speedups. Low-bit quantization (INT8) is another recent approach which improves decoding speed and reduces the memory footprint of models (Zafrir et al., 2019; Quinn and Ballesteros, 2018).

For model and architecture based speedups, we focus our attention on Transformer based NMT models which are now the work-horses in NLP and MT (Vaswani et al., 2017). While transformers are faster to train compared to their predecessors, Recurrent Neural Network (RNN) encoder-decoders (Bahdanau et al., 2014; Luong et al., 2015), transformers suffer from slower decoding speed. Subsequently, there has been interest in improving the decoding speed of transformers.

Shallow Decoders (SD): Shallow decoder models simply reduce the decoder depth and increase the encoder depth in response to the observation that decoding latency is proportional to the number of decoder layers (Kim et al., 2019; Miceli Barone et al., 2017; Wang et al., 2019; Kasai et al., 2020). Alternatively, one can employ SD models without increasing the encoder layers resulting in smaller (and faster) models.

Average Attention Networks (AAN): Average Attention Networks reduce the quadratic complexity of the decoder attention mechanism to linear time by replacing the decoder-side self-attention with an average-attention operation using a fixed weight for all time-steps (Zhang et al., 2018). This results in a $\approx$ 3-4x decoding speedup over the standard transformer.

4 Experimental Setup

Our objective is not to compare the various optimization methods against each other, but rather surface the impact of these algorithmic choices on gender biases. We treat all the optimization choices described in section 3 as data points available to conduct our analysis. To this end, we train models with all combinations of optimizations described in section 3 using the Fairseq toolkit (Ott et al., 2019). Our baseline is a standard large transformer with a $(6, 6)$ encoder-decoder layer configuration. For our SD models we use the following encoder-decoder layer configurations $\{(8, 4), (10, 2), (11, 1)\}$. We also train smaller shallow decoder (SSD) models without increasing the encoder depth $\{(6, 4), (6, 2), (6, 1)\}$. For each of these 7 configurations, we train AAN versions. Next, we save quantized and non-quantized versions for the 14 models, and decode with beam sizes of 1 and 5. We repeat our analysis for English to Spanish and English to German directions, using WMT13 En-Es and WMT14 En-De data sets, respectively. For the En-Es we limited the training data to $4M$ sentence pairs (picked at random without replacement) to ensure that the training for the two language directions have comparable data sizes. We apply Byte-Pair Encoding (BPE) with $32k$ merge operations to the data (Sennrich et al., 2016).

We measure decoding times and BLEU scores for the model's translations using the WMT test sets. Next, we evaluate each model's performance on SimpleGEN, specifically calculating the percent of correctly gendered nouns, incorrectly gendered nouns as well as inconclusive results. Table 3 shows an example of our evaluation protocol for an example source sentences and four possible translations. We deem the first two as correct even though the second translation incorrectly translates "funny" as "feliz" since we focus on the translation of "physician" only. The third translation is deemed incorrect because the masculine form "médico" is used and the last translation is deemed inconclusive since it is in the plural form. We average these metrics over 3 trials, each initialized with different random seeds. We obtained 56 data points for each language direction.

5 Analysis

Table 4a shows the performance of 6 selected models including a baseline transformer model with 6 encoder and decoder layers. The first two columns (time and BLEU) were computed using the WMT test sets. The remaining columns report metrics using SimpleGEN. The algorithmic choices resulting in the highest speed-up, result in a 1.5% and 4% relative drop in BLEU for En-Es and En-De, respectively (compared to the baseline model). The pro-stereotypical (pro) column shows the percentage correct gendered translation for sentences where the occupation gender matches the context gender. As expected the accuracies are relatively high (80.9 to 77.7) for all the models. The

direction	model	time(s)	BLEU	pro	anti	Δ	FOFC	MOFC	ΔFC	MOMC	FOMC	ΔMC
En-Es	baseline (bl)	3,662.8	33.2	80.9	44.2	36.7	69.4	41.7	27.7	88.2	48.1	40.0
	bl w/ bs=1	2,653.1	32.7	79.5	44.9	34.6	68.4	42.8	25.6	86.6	48.2	38.4
	bl w/ AAN	3,009.4	32.9	78.6	37.8	40.8	67.4	33.6	33.8	85.6	44.3	41.3
	bl w/ SD(10, 2)	2,241.7	32.9	77.9	38.1	39.8	67.3	35.9	31.4	84.6	41.7	42.9
	bl w/ SSD(6, 2)	1,993.5	32.7	77.7	38.7	39.0	66.0	33.8	32.2	85.1	46.3	38.8
	bl w/ quantization	2,116.1	32.7	79.8	41.4	38.4	67.0	37.2	29.8	88.0	48.1	39.8
	max rel. % drop	45.6	1.5	3.9	15.1		4.9	21.4		4.0	13.5	
En-De	baseline (bl)	3,653.0	27.2	67.7	39.7	28.0	57.5	31.6	25.9	74.2	52.3	21.8
	bl w/ bs=1	2,504.5	26.7	65.0	39.2	25.8	51.5	29.7	21.8	73.5	54.0	19.5
	bl w/ AAN	2,600.0	27.1	68.5	33.0	35.5	58.0	23.9	34.1	75.3	47.4	27.8
	bl w/ SD(10, 2)	1,960.8	27.1	67.5	32.6	35.0	57.7	26.5	31.2	73.8	46.7	27.1
	bl w/ SSD(6, 2)	2,091.0	27.0	66.9	35.9	31.0	56.6	30.3	26.2	73.5	44.6	28.9
	bl w/ quantization	2,205.1	26.1	63.2	33.2	30.0	50.5	24.6	25.9	71.3	46.8	24.6
	max rel. % drop	46.3	4.0	6.5	17.9		13.0	22.1		5.3	9.5	

(a) Each speed-up optimization individually.

direction	model	time(s)	BLEU	pro	anti	Δ	FOFC	MOFC	ΔFC	MOMC	FOMC	ΔMC
En-Es	baseline	3,662.8	33.2	80.9	44.2	36.7	69.4	41.7	27.7	88.2	48.1	40.0
	+bs=1	2,653.1	32.7	79.5	44.9	34.6	68.4	42.8	25.6	86.6	48.2	38.4
	+AAN	1,971.8	32.5	77.4	38.5	38.9	67.4	34.9	32.5	83.7	44.0	39.7
	+SD(10, 2)	1,164.2	32.1	75.3	36.2	39.1	57.1	31.7	25.3	86.8	43.2	43.6
	+SSD(6, 2)	1,165.7	31.9	78.6	40.4	38.2	66.9	36.3	30.5	86.0	46.8	39.2
	+quantization	679.6	31.1	73.1	34.9	38.2	58.7	29.5	29.2	82.3	43.4	38.8
	max rel. % drop	81.4	6.3	9.6	22.3		17.7	31.0		6.7	10.4	
En-De	baseline	3,653.0	27.2	67.7	39.7	28.0	57.5	31.6	25.9	74.2	52.3	21.8
	+bs=1	2,504.5	26.7	65.0	39.2	25.8	51.5	29.7	21.8	73.5	54.0	19.5
	+AAN	2,176.6	26.3	66.7	32.2	34.5	54.6	22.1	32.5	74.4	48.1	26.3
	+SD(10, 2)	1,332.3	25.8	64.2	29.1	35.1	50.3	22.2	28.1	73.0	44.7	28.3
	+SSD(6, 2)	1,153.2	25.7	64.7	28.9	35.9	53.9	19.9	34.1	71.6	43.0	28.6
	+quantization	732.6	24.7	61.0	23.3	37.6	46.3	14.8	31.5	70.3	36.7	33.6
	max rel. % drop	79.9	9.2	9.9	41.3		19.5	53.2		5.5	29.8	

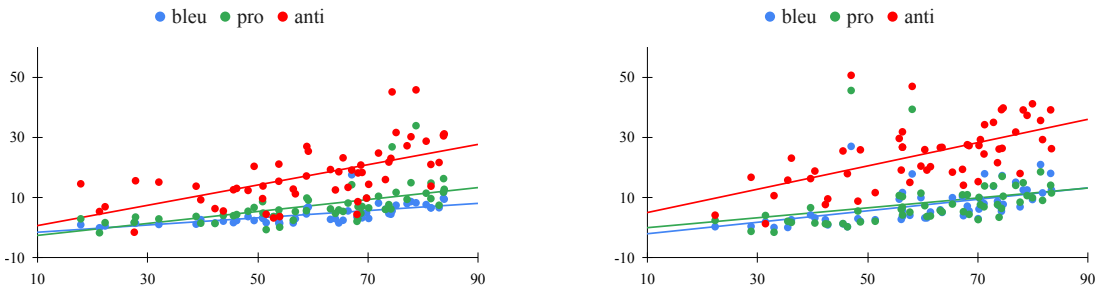
(b) "Stacked" speed-up optimizations.

Table 4: Results showing the effect of speed-up optimizations applied individually (in Table 4a) and stacked in Table 4b). We selected 6 models in both sections to highlight their effect on decoding time, BLEU and the % correctness on gender-bias metrics. The last row for each section (and each direction), shows the relative % drops in all the metrics between the fastest optimization method and the baseline. For example, for En-Es the relative % drop of decoding time for Table 4a is calculated as $100 * (3662.8 - 1993.5)/3662.8$.

last row in each section shows the *maximum relative drop* in each metric. We find that for the pro-stereotypical column the maximum relative drop is 1.5 and 6.5 for Spanish and German, respectively, which is similar to the relative change in BLEU scores. However, we find that the models are able to perform better on MOMC compared to FOFC suggesting biases even within the pro-stereotypical setting. In the anti-stereotypical (anti) column, we observe below-chance accuracies of only 44.2% and 39.7% for the two language directions, even from our best model. Columns FOFC and MOFC, show the difference in performance for sentences in the female context (FC) category in the presence of a stereotypical female occupation versus a stereotypical male occupation. We see a large imbalance in performance in these two columns summarized in ΔFC. Similarly, ΔMC summarizes the drop in performance when the model is confronted with stereotypical female occupations in a male context when compared to a male occupation in a male context. This suggests that the transformer's handling of grammatical agreement especially in cases where an occupation and contextual gender mismatch could be improved. The speedups disproportionately affect female context (FC) sentences across all categories.

In terms of model choices, we find that AANs deliver moderate speed-ups and minimal BLEU reduction compared to the baseline. However, AANs suffer the most degradation in terms of gender-bias. Δ, ΔFC and ΔMC are the highest for the ANN model in both language directions. On the other hand, greedy decoding with the baseline model has the smallest degradation in terms of gender-bias.

While Table 4a reveals the effect of select individual model choices, NMT practitioners, typically "stack" the optimization techniques together for large-scale deployment of NMT systems. Table 4b shows that stacking can provide $\approx 80 - 81\%$ relative drop in decoding time. However, we again see a disturbing trend where large speedups and small BLEU drops are accompanied with large drops in gender test performance. Again, FC sentences disproportionately suffer large drops in accuracy, particularly in MOFC in the En-De direction, where



(a) English-Spanish (b) English-German

Figure 1: Plots showing *relative percentage drop* of BLEU and gender-test metrics on the y-axis and *relative percentage drop* in decoding time in the x-axis FOr the two language directions analyzed. A breakdown of pro and anti into their constituent groups MOMC, FOFC, MOFc and FOMC is shown in Appendix A.3.

we see a 53.2% relative drop between the baseline and the fastest optimization stack.

While tables 4a and 4b show select models, we illustrate and further confirm our findings using all the data points (56 models trained) using scatter plots shown in fig. 1. We see that relative % drop in BLEU aligns closely with the relative % drop in gendered translation in the pro-stereotypical setting. In the case of German, the two trendlines are virtually overlapping. However, we see a steep drop for the anti-stereotypical settings, suggesting that BLEU scores computed using a typical test set only captures the stereotypical cases and even small reduction in BLEU could result in more instances of biased translations, especially in female context sentences.

6 Related Work

Previous research investigating gender bias in NMT has focused on data bias, ranging from assessment to mitigation. For example, Stanovsky et al. (2019b) adapted an evaluation data set for co-reference resolution to measure gender biases in machine translation. The sentences in this test set were created with ambiguous syntax, thus forcing the NMT model to "guess" the gender of the occupations. In contrast, there is always an unambiguous signal specifying the occupation-noun's gender in SimpleGEN. Similar work in speech-translation also studies contextual hints, but their work uses real-world sentences with complicated syntactic structures and sometimes the contextual hints are across sentence boundaries resulting in gender-ambiguous sentences (Bentivogli et al., 2020).

Zmigrod et al. (2019) create a counterfactual data-augmentation scheme by converting between masculine and feminine inflected sentences. Thus, with the additional modified sentences, the augmented data set equally represents both genders. Vanmassenhove et al. (2018), Stafanovičs et al. (2020) and Saunders et al. (2020) propose a data-annotation scheme in which the NMT model is trained to obey gender-specific tags provided with the source sentence. While Escudé Font and Costa-jussà (2019) employ pre-trained word-embeddings which have undergone a "debiasing" process (Bolukbasi et al., 2016; Zhao et al., 2018). Saunders and Byrne (2020) and Costa-jussà and de Jorge (2020) propose domain-adaptation on a carefully curated data set that "corrects" the model's misgendering problems. Costa-jussà et al. (2020) consider variations involving the amount of parameter-sharing between different language directions in multilingual NMT models.

7 Conclusion

With the current mainstreaming of machine translation, and its impact on people's everyday lives, bias mitigation in NMT should extend beyond data modifications and counter bias amplification due to algorithmic choices as well. We focus on algorithmic choices typically considered in speed-accuracy trade offs during productionization of NMT models. Our work illustrates that such trade offs, given current algorithmic choice practices, result in significant impact on gender translation, namely amplifying biases. In the process of this investigation, we construct a new gender translation evaluation set, SimpleGEN, and use it to show that modern NMT architectures struggle to overcome gender biases even when translating source sentences that are syntactically unambiguous and clearly marked for gender.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29, pages 4349–4357. Curran Associates, Inc.

Marta R Costa-jussà, Carlos Escolano, Christine Basta, Javier Ferrando, Roser Batlle, and Ksenia Kharitonova. 2020. Gender bias in multilingual neural machine translation: The architecture matters. *arXiv preprint arXiv:2012.13176*.

Marta R. Costa-jussà and Adrià de Jorge. 2020. Fine-tuning neural machine translation on gender-balanced datasets. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 26–34, Barcelona, Spain (Online). Association for Computational Linguistics.

Joel Escudé Font and Marta R. Costa-jussà. 2019. Equalizing gender bias in neural machine translation with word embeddings techniques. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 147–154, Florence, Italy. Association for Computational Linguistics.

Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 WMT shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 525–542, Berlin, Germany. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *arXiv preprint arXiv:2006.10369*.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 99–107, Copenhagen, Denmark. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jerry Quinn and Miguel Ballesteros. 2018. Pieces of eight: 8-bit neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 114–120, New Orleans - Louisiana. Association for Computational Linguistics.

Danielle Saunders and Bill Byrne. 2020. Reducing gender bias in neural machine translation as a domain adaptation problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online. Association for Computational Linguistics.

Danielle Saunders, Rosie Sallis, and Bill Byrne. 2020. Neural machine translation doesn't translate gender coreference right unless you make it. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 35–43, Barcelona, Spain (Online). Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Artūrs Stafanovičs, Mārcis Pinnis, and Toms Bergmanis. 2020. Mitigating gender bias in machine translation with target gender annotations. In *Proceedings of the Fifth Conference on Machine Translation*, pages 629–638, Online. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019a. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. 2019b. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

A Appendices

A.1 Impact Statement

This work identifies a weakness of NMT models where they appear to ignore contextual evidence regarding the gender of an occupation noun and apply an incorrect gender marker. It is difficult to measure the adverse effects of biases in NMT, but errors like the ones we highlight reduce trust in the NMT system.

Intended use: We hope that this type of error is further studied by NMT researchers leading to a solution. Furthermore, we expect the speed-optimization aspect of our work provides NMT engineers with an extra point of consideration, as we show gender-bias (errors in our dataset) increases rapidly compared to metrics like BLEU on standard datasets. In this work, we limit ourselves to viewing gender in the linguistic sense. SimpleGEN is not meant to be a replacement for traditional MT evaluation.

Risks: We recognize that socially, gendered language evolves (e.g. in English, "actress" is rarely used anymore). To the best of our knowledge, we selected occupations that are typically gendered (in Spanish and German) at present. Furthermore, we only regard the gender binary as a linguistic construct. It would be incorrect to use this work in the context of gender identity or gender expression etc.

Dataset: The dataset is "synthetic" in that it has been constructed using templates. We did not use crowd-sourcing or private data.

A.2 Full Template and Terms

Keywords	Values
f-n	female, women
m-n	male, men
f-n-pl	women, ladies, females, gals
m-n-pl	men, guys, males, fellows
f-n-sg	gal, woman, lady
m-n-sg	man, guy, fellow
f-obj-prn	her
m-obj-prn	him
f-pos-prn	her
m-pos-prn	his
f-obj-pos-prn	her
m-obj-pos-prn	his
f-sbj-prn	she
m-sbj-prn	he
f-rel	wife, mother, sister, girlfriend
m-rel	husband, father, brother, boyfriend

Table A5: Keywords and the values they can take.

Occupation Keywords	Values
f-occ-sg	clerk, designer, hairdresser, housekeeper, nanny, nurse, secretary
m-occ-sg	director, engineer, truck driver, farmer, laborer, mechanic, physician, president, plumber, carpenter, groundskeeper
f-occ-pl	clerks, designers, hairdressers, housekeepers, nannies, nurses, secretaries
m-occ-pl	directors, engineers, truck drivers, farmers, laborers, mechanics, physicians, presidents, plumbers, carpenters, groundskeepers
f-occ-sg-C	clerk, designer, hairdresser, housekeeper, nanny, nurse, secretary
m-occ-sg-C	director, truck driver, farmer, laborer, mechanic, physician, president, plumber, carpenter, groundskeeper
f-occ-pl-C	clerks, designers, hairdressers, housekeepers, nannies, nurses, secretaries
m-occ-pl-C	directors, truck drivers, farmers, laborers, mechanics, physicians, presidents, plumbers, carpenters, groundskeepers
f-occ-sg-V	
m-occ-sg-V	engineer,
f-occ-pl-V	
m-occ-pl-V	engineers,

Table A6: Occupation keywords and the values they can take. The prefix "m-" and "f-" indicate that according to the U.S Department of Labor these occupations have a higher percentage of male and female works, respectively.

Table A7 shows the template we use to generate our source sentences in SimpleGEN. We can generate sentences in one of the four sub-categories (MOMC, MOFC, FOFC, FOMC) by setting occupation keywords with the prefix m- or f- from our terminology set Table A6). For example, to generate MOFC sentences, we set occupation-keywords with prefix m- and non-occupation keywords with prefix f-.

A.3 Breakdown of scatter plots

Figures A2a and A2b further divides pro-stereotypical into male-occupations in male contexts (MoMc) and female-occupations in female context (FoFc), and anti-stereotypical into male-occupations in female contexts (MoFc) and female-occupations in male contexts (FoMc).

A.4 Evaluation Dictionary

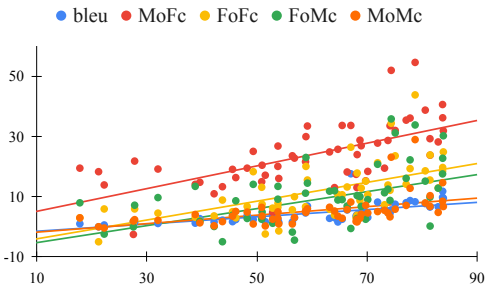
Table A8 shows the dictionary we use for evaluation.

My {f/m}-rel is a {f/m}-occ-sg-C .
My {f/m}-rel is an {f/m}-occ-sg-V .
His {f/m}-rel is a {f/m}-occ-sg-C .
His {f/m}-rel is an {f/m}-occ-sg-V .
Alex's {f/m}-rel became a {f/m}-occ-sg-C .
Alex's {f/m}-rel became an {f/m}-occ-sg-V .
My {f/m}-rel enjoys {f/m}-pos-prn work a {f/m}-occ-sg-C .
My {f/m}-rel enjoys {f/m}-pos-prn work an {f/m}-occ-sg-V .
Those {f/m}-n-pl are {f/m}-occ-pl .
These {f/m}-n-pl work hard {f/m}-occ-pl .
I talked to {f/m}-obj-prn , {f/m}-sbj-prn is a {f/m}-occ-sg-C .
I talked to {f/m}-obj-prn , {f/m}-sbj-prn is an {f/m}-occ-sg-V .
{f/m}-sbj-prn works a {f/m}-occ-sg-C .
{f/m}-sbj-prn works an {f/m}-occ-sg-V .
I worked with {f/m}-obj-prn , {f/m}-sbj-prn is a very qualified {f/m}-occ-sg .
{f/m}-n {f/m}-occ-pl are rare .
{f/m}-n {f/m}-occ-pl are not predisposed for self-promotion .
{f/m}-n {f/m}-occ-pl performed well .
{f/m}-n {f/m}-occ-pl earn less than their counterparts .
{f/m}-sbj-prn enjoys {f/m}-pos-prn work a {f/m}-occ-sg-C .
{f/m}-sbj-prn enjoys {f/m}-pos-prn work an {f/m}-occ-sg-V .
Though {f/m}-sbj-prn argued with {f/m}-pos-prn colleagues, the {f/m}-occ-sg was always respectful .
People laughed at the {f/m}-occ-sg behind {f/m}-obj-pos-prn back .
The {f/m}-occ-sg screamed when {f/m}-sbj-prn saw the mouse .
That {f/m}-occ-sg chooses {f/m}-obj-prn words very carefully .
That {f/m}-occ-sg is a funny {f/m}-n-sg !
Those {f/m}-occ-pl are talented {f/m}-n-pl !
Their {f/m}-occ-sg is a smart {f/m}-n-sg !
The {f/m}-occ-sg is a very dynamic {f/m}-n-sg !
The {f/m}-occ-sg is a {f/m}-n-sg with a lot of experience.
This {f/m}-occ-sg happens to be my {f/m}-rel .
The {f/m}-occ-sg works hard and {f/m}-sbj-prn excels !
The {f/m}-occ-sg drove fast and {f/m}-sbj-prn was on time .
As a {f/m}-occ-sg-C , {f/m}-sbj-prn is very competent .
As an {f/m}-occ-sg-V , {f/m}-sbj-prn is very competent .
When the {f/m}-occ-sg arrived, everyone applauded {f/m}-obj-prn .
The {f/m}-occ-pl with the highest success rate per the polls, were {f/m}-n .
That {f/m}-occ-sg is also my {f/m}-rel .
The {f/m}-occ-sg moved here from Ethiopia where {f/m}-sbj-prn is from .
The {f/m}-occ-sg arrives early for {f/m}-pos-prn shift .
That {f/m}-occ-sg is hella strict, watch out for {f/m}-obj-prn .
That {f/m}-occ-sg retired early, good for {f/m}-obj-prn .

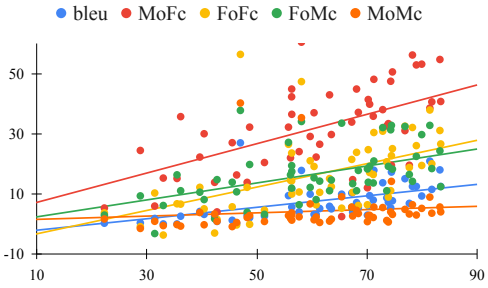
Table A7: Our template set used to generate English source sentences.

English	Spanish-Masc	Spanish-Fem	German-Masc	German-Fem
clerk	funcionario|empleado administrativo	funcionaria|empleada administrativa	Mitarbeiter	Mitarbeiterin
designer	diseñador	diseñadora	Designer	Designerin
hairdresser	peluquero	peluquera	Friseur	Friseurin
housekeeper	amo de llaves|amo de casa	ama de llaves|ama de casa	Haushälter	Haushälterin
nanny	niñero	niñera	Kinderbetreuer	Kindermädchen
nurse	enfermero	enfermera	Krankenpfleger	Krankenschwester
secretary	secretario	secretaria	Sekretär	Sekretärin
director	director	directora	Direktor	Direktorin
engineer	ingeniero	ingeniera	Ingenieur	Ingenieurin
truck driver	camionero	camionera	Lastwagenfahrer	Lastwagenfahrerin
farmer	granjero|campesino	granjera|campesina	Bauer	Bäuerin
laborer	obrero|trabajador	obrera|trabajadora	Arbeiter	Arbeiterin
mechanic	mecánico	mecánica	Mechaniker	Mechanikerin
physician	médico	médica	Arzt	Ärztin
president	presidente	presidenta	Präsident	Präsidentin
plumber	plomero|fontanero	plomera|fontanera	Klempner	Klempnerin
carpenter	carpintero	carpintera	Tischler|Zimmermann	Tischlerin
groundskeeper	jardinero|guardián	jardinera|guardiana	Gärtner	Gärtnerin
clerks	funcionarios	funcionarias	MItarbeiter	Mitarbeiterinnen
designers	diseñadores	diseñadoras	Designer	Designerinnen
hairdressers	peluqueros	peluqueras	Friseure	Friseurinnen
housekeepers	amos de llaves|amos de casa	amas de llaves|amas de casa	Haushälter	Haushälterinnen
nannies	niñeros	niñeras	Kinderbetreuer	Kindermädchen
nurses	enfermeros	enfermeras	Krankenpfleger	Krankenschwestern
secretaries	secretarios	secretarias	Sekretäre	Sekretärinnen
directors	directores	directoras	Direktoren	Direktorinnen
engineers	ingenieros	ingenieras	Ingenieuren	Ingenieurinnin
truck drivers	camioneros	camioneras	Lastwagenfahrerin	Lastwagenfahrerinnen
farmers	granjeros	granjeras	Bauern	Bäuerinnen
laborers	obreros	obreras	Arbeiter	Arbeiterinnen
mechanics	mecánicas	mecánicos	Mechaniker	Mechanikerinnen
physicians	médico	médicas	Ärzte	Ärztinnen
presidents	presidentes	presidentas	Präsidenten	Präsidentinnen
plumbers	plomeros	plomeras	Klempner	Klempnerinnen
carpenters	carpinteros	carpinteras	Tischler	Tischlerinnen
groundskeepers	jardineros|guardianes	jardineras|guardianas	Gärtner	Gärtnerinnen

Table A8: Our dictionary of occupations. Entries with the "|" symbol indicate that we accept either of the references as correct.



(a) English-Spanish



(b) English-German

Figure A2: Plots showing *relative percentage drop* of BLEU and gender-test metrics on the y-axis and *relative percentage drop* in decoding time in the x-axis.

Machine Translation into Low-resource Language Varieties

Sachin Kumar[♣] **Antonios Anastasopoulos**[♢] **Shuly Wintner**[♡] **Yulia Tsvetkov**[♠]
[♣]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[♢]Department of Computer Science, George Mason University, Fairfax, VA, USA
[♡]Department of Computer Science, University of Haifa, Haifa, Israel
[♠]Paul G. Allen School of Computer Science & Engineering, University of Washington
sachink@cs.cmu.edu, antonis@gmu.edu, shuly@cs.haifa.ac.il, yuliats@cs.washington.edu

Abstract

State-of-the-art machine translation (MT) systems are typically trained to generate "standard" target language; however, many languages have multiple varieties (regional varieties, dialects, sociolects, non-native varieties) that are different from the standard language. Such varieties are often low-resource, and hence do not benefit from contemporary NLP solutions, MT included. We propose a general framework to rapidly adapt MT systems to generate language varieties that are close to, but different from, the standard target language, using no parallel (source–variety) data. This also includes adaptation of MT systems to low-resource typologically-related target languages.[1] We experiment with adapting an English–Russian MT system to generate Ukrainian and Belarusian, an English–Norwegian Bokmål system to generate Nynorsk, and an English–Arabic system to generate four Arabic dialects, obtaining significant improvements over competitive baselines.

1 Introduction

Despite tremendous progress in machine translation (Bahdanau et al., 2015; Vaswani et al., 2017) and language generation in general, current state-of-the-art systems often work under the assumption that a language is homogeneously spoken and understood by its speakers: they generate a "standard" form of the target language, typically based on the availability of parallel data. But language use varies with regions, socio-economic backgrounds, ethnicity, and fluency, and many widely spoken languages consist of dozens of varieties or dialects, with differing lexical, morphological, and syntactic patterns for which no translation data are typically available. As a result, models trained to translate from a source language (SRC) to a standard language variety (STD) lead to a sub-par experience for speakers of other varieties.

Motivated by these issues, we focus on the task of adapting a trained SRC→STD translation model to generate text in a different target variety (TGT), having access only to limited monolingual corpora in TGT and no SRC–TGT parallel data. TGT may be a dialect of, a language variety of, or a typologically-related language to STD.

We present an effective transfer-learning framework for translation into low resource language varieties. Our method reuses SRC→STD MT models and finetunes them on synthesized (pseudo-parallel) SRC–TGT texts. This allows for rapid adaptation of MT models to new varieties without having to train everything from scratch. Using word-embedding adaptation techniques, we show that MT models which predict continuous word vectors (Kumar and Tsvetkov, 2019) rather than softmax probabilities lead to superior performance since they allow additional knowledge to be injected into the models through transfer between word embeddings of high-resource (STD) and low-resource (TGT) monolingual corpora.

We evaluate our framework on three translation tasks: English to Ukrainian and Belarusian, assuming parallel data are only available for English→Russian; English to Nynorsk, with only English to Norwegian Bokmål parallel data; and English to four Arabic dialects, with only English→Modern Standard Arabic (MSA) parallel data. Our approach outperforms competitive baselines based on unsupervised MT, and methods based on finetuning softmax-based models.

2 A Transfer-learning Architecture

We first formalize the task setup. We are given a parallel SRC→STD corpus, which allows us to

[1]Code, data and trained models are available here: https://github.com/Sachin19/seq2seq-con

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 110–121
August 1–6, 2021. ©2021 Association for Computational Linguistics

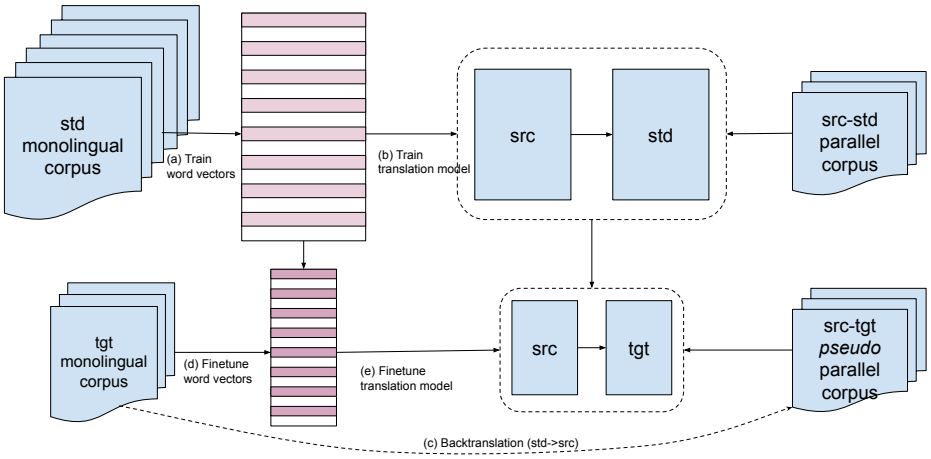


Figure 1: An overview of our approach. (a) Using the available STD monolingual corpora, we first train word vectors using `fasttext`; (b) we then train a SRC→STD translation model using the parallel corpora to predict the pretrained word vectors; (c) next, we train STD→SRC model and use it to translate TGT monolingual corpora to SRC; (d) now, we finetune STD subword embeddings to learn TGT word embeddings; and finally (e) we finetune a SRC→STD model to generate TGT pretrained embeddings using the back-translated SRC→TGT data.

train a translation model $f(\cdot;\theta)$ that takes an input sentence x in SRC and generates its translation in the standard veriety STD, $\hat{y}_{\text{STD}} = f(x;\theta)$. Here, θ are the learnable parameters of the model. We are also given monolingual corpora in both the standard STD and target variety TGT. Our goal now is to modify f to generate translations $\hat{y}_{\text{TGT}}$ in the target variety TGT. At training time, we assume no SRC–TGT or STD–TGT parallel data are available.

Our solution (Figure 1) is based on a transformer-based encoder-decoder architecture (Vaswani et al., 2017) which we modify to predict word vectors. Following Kumar and Tsvetkov (2019), instead of treating each token in the vocabulary as a discrete unit, we represent it using a unit-normalized d-dimensional pre-trained vector. These vectors are learned from a STD monolingual corpus using `fasttext` (Bojanowski et al., 2017). A word's representation is computed as the average of the vectors of its character n-grams, allowing surface-level linguistic information to be shared among words. At each step in the decoder, we feed this pretrained vector at the input and instead of predicting a probability distribution over the vocabulary using a softmax layer, we predict a d-dimensional continuous-valued vector. We train this model by minimizing the von Mises-Fisher (vMF) loss—a probabilistic variant of cosine distance—between the predicted vector and the pre-trained vector. The pre-trained vectors (at both input and output of the decoder) are not trained with the model. To decode from this model, at each step, the output word is generated by finding the closest neighbor (in terms of cosine similarity) of the predicted output vector in the pre-trained embedding table.

We train f in this fashion using SRC–STD parallel data. As shown below, training a softmax-based SRC→STD model to later finetune with TGT suffers from vocabulary mismatch between STD and TGT and thus is detrimental to downstream performance. By replacing the decoder input and output with pre-trained vectors, we separate the vocabulary from the MT model, making adaptation easier.

Now, to finetune this model to generate TGT, we need TGT embeddings. Since the TGT monolingual corpus is small, training `fasttext` vectors on this corpus from scratch will lead (as we show) to low-quality embeddings. Leveraging the relatedness of STD and TGT and their vocabulary overlap, we use STD embeddings to transfer knowledge to TGT embeddings: for each character n-gram in the TGT corpus, we initialize its embedding with the corresponding STD embedding, if available. We then continue training `fasttext` on the TGT monolingual corpus (Chaudhary et al., 2018). Last, we use a supervised embedding alignment method (Lample et al., 2018a) to project the learned TGT embeddings in the same space as STD. STD and TGT are expected to have a large lexical overlap, so we use identical tokens in both varieties as supervision for this alignment. The obtained embeddings, due to transfer learning from STD, inject additional knowledge in the model.

Finally, to obtain a SRC→TGT model, we finetune f on psuedo-parallel SRC–TGT data. Using a STD→SRC MT model (a back-translation model

trained using large STD–SRC parallel data with standard settings) we (back)-translate TGT data to SRC. Naturally, these synthetic parallel data will be noisy despite the similarity between STD and TGT, but we show that they improve the overall performance. We discuss the implications of this noise in §4.

3 Experimental Setup

Datasets We experiment with two setups. In the first (synthetic) setup, we use English (EN) as SRC, Russian (RU) as STD, and Ukrainian (UK) and Belarusian (BE) as TGTs. We sample 10M EN-RU sentences from the WMT'19 shared task (Ma et al., 2019), and 80M RU sentences from the CoNLL'17 shared task to train embeddings. To simulate low-resource scenarios, we sample 10K, 100K and 1M UK sentences from the CoNLL'17 shared task and BE sentences from the OSCAR corpus (Ortiz Suárez et al., 2020). We use TED dev/test sets for both languages pairs (Cettolo et al., 2012).

The second (real world) setup has two language sets: the first one defines English as SRC, with Modern Standard Arabic (MSA) as STD and four Arabic varieties spoken in Doha, Beirut, Rabat and Tunis as TGTs. We sample 10M EN-MSA sentences from the UNPC corpus (Ziemski et al., 2016), and 80M MSA sentences from the CoNLL'17 shared task. For Arabic varieties, we use the MADAR corpus (Bouamor et al., 2018) which consists of 12K 6-way parallel sentences between English, MSA and the 4 considered varieties. We ignore the English sentences, sample dev/test sets of 1K sentences each, and consider 10K monolingual sentences for each TGT variety. The second set also has English as SRC with Norwegian Bokmål (NO) as STD and its written variety Nynorsk (NN) as TGT. We use 630K EN-NO sentences from WikiMatrix (Schwenk et al., 2021), and 26M NO sentences from ParaCrawl (Esplà et al., 2019) combined with the WikiMatrix NO sentences to train embeddings. We use 310K NN sentences from WikiMatrix, and TED dev/test sets for both varieties (Reimers and Gurevych, 2020).

Preprocessing We preprocess raw text using Byte Pair Encoding (BPE, Sennrich et al., 2016) with 24K merge operations on each SRC–STD corpus trained separately on SRC and STD. We use the same BPE model to tokenize the monolingual STD data and learn `fasttext` embeddings (we consider character n-grams of length 3 to 6).[2] Splitting the TGT words with the same STD BPE model will result in heavy segmentation, especially when TGT contains characters not present in STD.[3] To counter this, we train a joint BPE model with 24K operations on the concatenation of STD and TGT corpora to tokenize TGT corpus following Chronopoulou et al. (2020). This technique increases the number of shared tokens between STD and TGT, thus enabling better cross-variety transfer while learning embeddings *and* while finetuning. We follow Chaudhary et al. (2018) to train embeddings on the generated TGT vocabulary where we initialize the character n-gram representations for TGT words with STD's `fasttext` model wherever available and finetune them on the TGT corpus.

Implementation and Evaluation We modify the standard OpenNMT-py seq2seq models of PyTorch (Klein et al., 2017) to train our model with vMF loss (Kumar and Tsvetkov, 2019). Additional hyperparameter details are outlined in Appendix B. We evaluate our methods using BLEU score (Papineni et al., 2002) based on the SacreBLEU implementation (Post, 2018).[4] For the Arabic varieties, we also report a macro-average. In addition, to measure the expected impact on actual systems' users, we follow Faisal et al. (2021) in computing a population-weighted macro-average ($\textbf{avg}_{\textbf{pop}}$) based on language community populations provided by Ethnologue (Eberhard et al., 2019).

3.1 Experiments

Our proposed framework, **LANGVARMT**, consists of three main components: (1) A supervised SRC→STD model is trained to predict continuous STD word embeddings rather than discrete softmax probabilities. (2) Output STD embeddings are replaced with TGT embeddings. The TGT embeddings are trained by finetuning STD embeddings on monolingual TGT data and aligning the two embedding spaces. (3) The resulting model is finetuned with pseudo-parallel SRC→TGT data.

We compare LANGVARMT with the following competitive baselines. **SUP(SRC→STD)**: train a standard (softmax-based) supervised SRC→STD model, and consider the output of this model as

[2] We slightly modify `fasttext` to not consider BPE token markers "@@" in the character n-grams.

[3] For example, both RU and UK alphabets consist of 33 letters; RU has the letters Ёё, ъ, ы and Ээ, which are not used in UK. Instead, UK has Ґґ, Єє, Ii and Її.

[4] While we recognize the limitations of BLEU (Mathur et al., 2020), more sophisticated embedding-based metrics for MT evaluation (Zhang et al., 2020; Sellam et al., 2020) are unfortunately not available for low-resource language varieties.

Size of TGT corpus	UK 10K	UK 100K	UK 1M	BE 10K	BE 100K	BE 1M	NN 300K	Arabic Varieties (10K) Doha	Beirut	Rabat	Tunis
SUP(SRC→STD)	1.7	1.7	1.7	1.5	1.5	1.5	11.3	3.7	1.8	2.0	1.3
UNSUP(SRC→TGT)	0.3	0.6	0.9	0.4	0.6	1.4	2.7	0.2	0.1	0.1	0.1
PIVOT	1.5	8.6	14.9	1.15	3.9	8.0	11.9	1.8	2.1	1.7	1.1
SOFTMAX	1.9	12.7	15.4	1.5	4.5	7.9	14.4	14.5	7.4	4.9	3.9
LANGVARMT	**6.1**	**13.5**	**15.3**	**2.3**	**8.8**	**9.8**	**16.6**	**20.1**	**8.1**	**7.4**	**4.6**

Table 1: BLEU scores on translation from English to Ukrainian, Belarusian, Nynorsk, and Arabic dialects with varying amounts of monolingual target data (TGT sentences) available for finetuning. Our approach (LANGVARMT) outperforms all baselines.

TGT under the assumption that STD and TGT may be very similar. **UNSUP(SRC→TGT)**: train an unsupervised MT model (Lample et al., 2018a) in which the encoder and decoder are initialized with cross-lingual masked language models (MLM, Conneau and Lample, 2019). These MLMs are pre-trained on SRC monolingual data, and then finetuned on TGT monolingual data with an expanded vocabulary as described above. This baseline is taken from Chronopoulou et al. (2020), where it showed state-of-the-art performance for low-monolingual-resource scenarios. **Pivot**: train a UNSUP(STD→TGT) model as described above using STD and TGT monolingual corpora. During inference, translate the SRC sentence to STD with the SUP(SRC→STD) model and then to TGT with the UNSUP(STD→TGT) model. We also perform several ablation experiments, showing that every component of LANGVARMT is necessary for good downstream performance. Specifically, we report results with LANGVARMT but using a standard softmax layer (**SOFTMAX**) to predict tokens instead of continuous vectors.[5]

4 Results and Analysis

Table 1 compares the performance of LANGVARMT with the baselines for Ukrainian, Belarusian, Nynorsk, and the four Arabic varieties. For reference, note that the EN→RU, EN→MSA, and EN→NO models are relatively strong, yielding BLEU scores of 24.3, 21.2, and 24.9, respectively.

Synthetic Setup Considering STD and TGT as the same language is sub-optimal, as is evident from the poor performance of the non-adapted SUP(SRC→STD) model. Clearly, special attention ought to be paid to language varieties. Direct unsupervised translation from SRC to TGT performs poorly as well, confirming previously reported results of the ineffectiveness of such methods on unrelated languages (Guzmán et al., 2019).

[5]Additional ablation results are listed in Appendix C.

Translating SRC to TGT by pivoting through STD achieves much better performance owing to strong UNSUP(STD→TGT) models that leverage the similarities between STD and TGT. However, when resources are scarse (e.g., with 10K monolingual sentences as opposed to 1M), this performance gain considerably diminishes. We attribute this drop to overfitting during the pre-training phase on the small TGT monolingual data. Ablation results (Appendix C) also show that in such low-resource settings the learned embeddings are of low quality.

Finally, LANGVARMT consistently outperforms all baselines. Using 1M UK sentences, it achieves similar performance (for EN→UK) to the softmax ablation of our method, SOFTMAX, and small gains over unsupervised methods. However, in lower resource settings our approach is clearly better than the strongest baselines by over 4 BLEU points for UK (10K) and 3.9 points for BE (100K).

To identify potential sources of error in our proposed method, we lemmatize the generated translations and test sets and evaluate BLEU (Qi et al., 2020). Across all data sizes, both UK and BE achieve a substantial increase in BLEU (up to +6 BLEU; see Appendix D for details) compared to that obtained on raw text, indicating morphological errors in the translations. In future work, we will investigate whether we can alleviate this issue by considering TGT embeddings based on morphological features of tokens (Chaudhary et al., 2018).

Real-world Setup The effectiveness of LANGVARMT is pronounced in this setup with a dramatic improvement of more than 18 BLEU points over unsupervised baselines when translating into Doha Arabic. We hypothesize that during the pretraining phase of unsupervised methods, the extreme difference between the size of the MSA monolingual corpus (10M) and the varieties' corpora (10K) leads to overfitting. Additionally, compared to the synthetic setup, the Arabic varieties we consider are quite close to MSA, allowing for easy and effective adaptation of both word embeddings and

EN→MSA models. LANGVARMT also improves in all other Arabic varieties, although naturally some varieties remain challenging. For example, the Rabat and particularly the Tunis varieties are more likely to include French loanwords (Bouamor et al., 2018) which are not adequately handled as they are not part of our vocabulary. In future work, we will investigate whether we can alleviate this issue by potentially including French corpora (transliterated into Arabic) to our TGT language corpora. On average, our approach improves by 2.3 BLEU points over the softmax-based baseline (cf. 7.7 and 10.0 in Table 2 under $\mathbf{avg}_{\mathcal{L}}$) across the four Arabic dialects. For a population-weighted average ($\mathbf{avg}_{\mathbf{pop}}$), we associate the Doha variety with Gulf Arabic (ISO code: `afb`), the Beirut one with North Levantine Arabic (`apc`), Rabat with Moroccan (`ary`), and the Tunis variety with Tunisian Arabic (`aeb`). As before, LANGVARMT outperforms the baselines. The absolute BLEU scores in this highly challenging setup are admittedly low, but as we discuss in Appendix D, the translations generated by LANGVARMT are often fluent and input preserving, especially compared to the baselines.

Finally, due to high similarity between NO and NN, the SUP(EN→NO) model also performs well on NN with 11.3 BLEU, but our method yields further gains of over 4 points over the baselines.

5 Discussion

Fairness The goal of this work is to develop more equitable technologies, usable by speakers of diverse language varieties. Here, we evaluate the systems along the principles of *fairness*. We evaluate the fairness of our Arabic multi-dialect system's utility proportionally to the populations speaking those dialects. In particular, we seek to measure how much average benefit will the people of different dialects receive if their respective translation performance is improved. A simple proxy for fairness is the standard deviation (or, even simpler, a $\max - \min$ performance) of the BLEU scores across dialects (A higher value implies more unfairness across the dialects) Beyond that, we measure a system's *unfairness* with respect to the different dialect subgroups, using the adaptation of generalized entropy index (Speicher et al., 2018), which considers equities within and between subgroups in evaluating the overall unfairness of an algorithm on a population Faisal et al. (2021) (See Appendix F for details and additional discussion).

Table 2 shows that our proposed method is fairer across all dialects, compared to baselines where only MSA translation produces comprehensible outputs.

Model	$\mathbf{avg}_{\mathcal{L}}\uparrow$	$\mathbf{avg}_{\mathbf{pop}}\uparrow$	max−min↓	unfair↓
SUP(SRC→STD)	2.2	1.8	19.9	0.037
UNSUP(SRC→TGT)	0.1	0.1	21.1	0.046
PIVOT	1.7	1.8	20.1	0.037
SOFTMAX	7.7	5.7	17.3	0.020
LANGVARMT	**10.0**	**7.3**	**16.6**	**0.016**

Table 2: Average performance and fairness metrics across the four Arabic varieties. This evaluation includes MSA (with a BLEU score of 21.2 on the SUP(EN→MSA) model).

Negative Results Our proposed method relies on two components: (1) quality of TGT word embeddings which is dependent on STD and TGT shared (subword) vocabulary, and (2) the psuedo-parallel SRC–TGT obtained by back-translating TGT data through a STD→SRC model. If STD and TGT are not sufficiently closely related, the quality of both of these components can degrade, leading to a drop in the performance of our proposed method. We present results of two additional experiments to elucidate this phenomenon in Appendix E.

Related Work We provide an extensive discussion of related work in Appendix A.

6 Conclusion

We presented a transfer-learning framework for rapid and effective adaptation of MT models to different varieties of the target language without access to any source-to-variety parallel data. We demonstrated significant gains in BLEU scores across several language pairs, especially in highly resource-scarce scenarios. The improvements are mainly due to the benefits of continuous-output models over softmax-based generation. Our analysis highlights the importance of addressing morphological differences between language varieties, which will be in the focus of our future work.

Acknowledgements

This research was supported by Grants No. 2017699 and 2019785 from the United States-Israel Binational Science Foundation (BSF), by the National Science Foundation (NSF) under Grants No. 2040926 and 2007960, and by a Google faculty research award. We thank Safaa Shehadi for evaluating our model outputs, Xinyi Wang and Aditi Choudhary for helpful discussions, and the anonymous reviewers for much appreciated feedback.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Kemal Altintas and Ilyas Cicekli. 2002. A machine translation system between a pair of closely related languages. In *In Seventeenth International Symposium On Computer and Information Sciences*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R. Mortensen, and Jaime Carbonell. 2018. Adapting word embeddings to new languages with morphological and phonological subword representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3285–3295, Brussels, Belgium. Association for Computational Linguistics.

Monojit Choudhury and Amit Deshpande. 2021. How linguistically fair are multilingual pre-trained language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, Online. AAAI.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2020. Reusing a Pretrained Language Model on Languages with Limited Corpora for Unsupervised NMT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2703–2711, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

David M Eberhard, Gary F Simons, and Charles D. (eds.) Fennig. 2019. Ethnologue: Languages of the world. 2019. *online. Dallas, Texas: SIL International.*

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Fahim Faisal, Sharlina Keshava, Md Mahfuz ibn Alam, and Antonios Anastasopoulos. 2021. SD-QA: Spoken Dialectal Question Answering for the Real World. Preprint.

Xavier Garcia, Pierre Foret, Thibault Sellam, and Ankur Parikh. 2020. A multilingual view of unsupervised machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3160–3170, Online. Association for Computational Linguistics.

Xavier Garcia, Aditya Siddhant, Orhan Firat, and Ankur Parikh. 2021. Harnessing multilinguality in unsupervised machine translation for rare languages. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1126–1137, Online. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*

and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Hieu Hoang and Philipp Koehn. 2008. Design of the Moses decoder for statistical machine translation. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 58–65, Columbus, Ohio. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Lidia Kidane, Sachin Kumar, and Yulia Tsvetkov. 2021. An exploration of data augmentation techniques for improving English to Tigrinya translation. In *Proceedings of the Second AfricaNLP Workshop*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Sachin Kumar and Yulia Tsvetkov. 2019. Von mises-fisher loss for training sequence to sequence models with continuous outputs. In *International Conference on Learning Representations*.

Surafel Melaku Lakew, Aliia Erofeeva, and Marcello Federico. 2018. Neural machine translation into language varieties. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 156–164, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *International Conference on Learning Representations*.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Kelly Marchisio, Kevin Duh, and Philipp Koehn. 2020. When does unsupervised machine translation work? In *Proceedings of the Fifth Conference on Machine Translation*, pages 571–583, Online. Association for Computational Linguistics.

Luis Marujo, Nuno Grazina, Tiago Luis, Wang Ling, Luisa Coheur, and Isabel Trancoso. 2011. BP2EP - adaptation of Brazilian Portuguese texts to European Portuguese. In *Proceedings of the 15th Annual conference of the European Association for Machine Translation*, Leuven, Belgium. European Association for Machine Translation.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of the*

2017 Conference on Empirical Methods in Natural Language Processing, pages 2513–2518, Copenhagen, Denmark. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

John Rawls. 1999. *A Theory of Justice*. Harvard University Press.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. 2018. A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2239–2248.

T. Tan, S. Goh, and Y. Khaw. 2012. A malay dialect translation and synthesis system: Proposal and preliminary system. In *2012 International Conference on Asian Language Processing*, pages 109–112.

Jörg Tiedemann. 2009. Character-based PSMT for closely related languages. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

David Vilar, Jan-T. Peter, and Hermann Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 33–39, USA. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

A Related Work

Early work addressing translation involving language varieties includes rule-based transformations (Altintas and Cicekli, 2002; Marujo et al., 2011; Tan et al., 2012) which rely on language specific information and expert knowledge which can be expensive and difficult to scale. Recent work to address this issue only focuses on cases where parallel data do exist. They include a combination of word-level and character-level MT (Vilar et al., 2007; Tiedemann, 2009; Nakov and Tiedemann, 2012) between related languages or training multilingual models to translate to/from English to different varieties of a language (e.g., Lakew et al. (2018) work on Brazilian–European Portuguese and European–Canadian French). Such parallel data, however, are typically unavailable for most language varieties.

Unsupervised translation models, which require only monolingual data, can address this limitation (Artetxe et al., 2018; Lample et al., 2018a; Garcia et al., 2020, 2021). However, when even *mono*lingual corpora are limited, unsupervised models are challenging to train and are quite ineffective for translating between unrelated languages (Marchisio et al., 2020). Considering varieties of a language as writing styles, unsupervised style transfer (Yang et al., 2018; He et al., 2020) or deciphering methods (Pourdamghani and Knight, 2017) to translate between different varieties have also been been explored but have not been shown to perform well, often only reporting BLEU-1 scores since they obtain BLEU-4 scores which are closer to 0. Additionally, all of these approaches require simultaneous access to data in all varieties during training and must be trained from scratch when a new variety is added. In contrast, our presented method allows for easy adaptation of SRC→STD models to any new variety as it arrives.

Considering a new target variety as a new domain of STD, unsupervised domain adaptation methods can be employed, such as finetuning SRC→STD models using pseudo-parallel corpora generated from monolingual corpora in target varieties (Hu et al., 2019; Currey et al., 2017). Our proposed method is most related to this approach; but while these methods have the potential to adapt the decoder language model, for effective transfer, STD and TGT must have a shared vocabulary which is not true for most language varieties due to lexical, morphological, and at times orthographic differences. In contrast, our proposed method makes use of cross-variety word embeddings. While our examples only involve same-script varieties, augmenting our approach to work across scripts through a transliteration component is straightforward.

B Implementation Details

We modify the standard OpenNMT-py seq2seq models of PyTorch (Klein et al., 2017) to train our model with vMF loss (Kumar and Tsvetkov, 2019). We use the transformer-BASE model (Vaswani et al., 2017), with 6 layers in both encoder and decoder and with 8 attention heads, as our underlying architecture. We modify this model to predict pretrained `fasttext` vectors. We also initialize the decoder input embedding table with the pretrained vectors and do not update them during model training. All models are optimized using Rectified Adam (Liu et al., 2020) with a batch size of 4K tokens and dropout of 0.1. We train SRC→STD models for 350K steps with an initial learning rate of 0.0007 with linear decay. For finetuning, we reduce the learning rate to 0.0001 and train for up to 100K steps. We use early stopping in all models based on validation loss computed every 2K steps. We decode all the softmax-based models with a beam size of 5 and all the vMF-based models greedily.

We evaluate our methods using BLEU score (Papineni et al., 2002) based on the SacreBLEU implementation (Post, 2018). While we recognize the limitations of BLEU (Mathur et al., 2020), more sophisticated embedding-based metrics for MT evaluation (Zhang et al., 2020; Sellam et al., 2020) are simply not available for language varieties.

C Additional English-Ukrainian Experiments

On our resource-richest setup of EN→UK translation using 1M UK sentences and RU as STD, we compare our method with the following additional baselines. Table 3 presents these results.

LAMPLE-UNSUP(SRC→TGT): This is another unsupervised model, based on Lample et al. (2018a) which initializes the input and output embedding tables of both encoder and decoder using cross-lingual word embeddings trained on SRC and TGT monolingual corpora. The model is trained in a similar manner to Chronopoulou et al. (2020) (UNSUP(SRC→TGT)) with iterative backtranslation and autoencoding.

PIVOT:LAMPLE(STD→TGT): This baseline is

Method	BLEU (uk)
SUP(SRC-STD)	1.7
UNSUP(SRC→TGT)	0.9
PIVOT:	14.9
LAMPLE-UNSUP(SRC→TGT)	0.4
PIVOT:LAMPLE-UNSUP(STD→TGT)	9.0
PIVOT:DICTREPLACE(STD→TGT)	2.9
LANGVARMT	15.3
LANGVARMT w/ poor embeddings	4.6
LANGVARMT-RANDOM	13.1
SOFTMAX	15.4
LANGVARMT-RANDOM-SOFTMAX	14.1

Table 3: BLEU scores on EN-UK test corpus with 1M UK monolingual corpus.

similar to the PIVOT baseline, where we replace the unsupervised model with that of Lample et al. (2018a).

PIVOT:DICTREPLACE(STD→TGT): Here we first translate SRC to STD using SUP(SRC→STD), and then modify the STD output to get a TGT sentence as follows: We create a STD–TGT dictionary using the embedding map suggested by Lample et al. (2018b). This dictionary is created on words tokenized with Moses tokenizer (Hoang and Koehn, 2008) rather than BPE tokens. We replace each token in the generated STD sentence which is not in the TGT vocabulary using the dictionary (if available). We consider this baseline to measure lexical vs. syntactic/phrase level differences between Russian and Ukrainian.

In addition to baseline comparison, we report the following ablation experiments.

(1) To measure transfer from STD to TGT embeddings, we finetune the SUP(SRC→STD) model using TGT embeddings trained from scratch (as opposed to initialized with STD embeddings).

(2) To measure the impact of initialization during model finetuning, we compare with a randomly initialized model trained in a supervised fashion on the psuedo-parallel SRC–TGT data.

Baselines On the unsupervised models based on Lample et al. (2018a), we observe a similar trend as that of Chronopoulou et al. (2020), where the LAMPLE-UNSUP(SRC→TGT) model performing poorly (0.4) with substantial gains when pivoting through Russian (9.0 BLEU).

PIVOT:DICTREPLACE(STD→TGT) gains some improvement over considering the output of SUP(SRC→STD) as TGT, probably due to syntactic similarities between Russian and Ukrainian. This result can potentially be further improved with a human-curated RU–UK dictionary, but such resources are typically not available for the low-resource settings we consider in this paper.

Ablations As shown in Table 3, training the SRC→TGT model on a randomly initialized model (LANGVAR-RANDOM) results in a performance drop, confirming that transfer learning from a SRC→STD model is beneficial. Similarly, using TGT embeddings trained from scratch (LANGVARMT w/ poor embeddings) results in a drastic performance drop, providing evidence for essential transfer from STD embeddings.

D Analysis

To better understand the performance of our models, we perform additional analyses.

Lemmatized BLEU For UK and BE, we lemmatize each word in the test sets and the translations and evaluate BLEU scores. The results, depicted in Table 4, very likely indicate that our framework often generates correct lemmas, but may fail on the correct inflectional form of the target words. This highlights the importance of considering morphological differences between language varieties. The high BLEU scores also demonstrate that the resulting translations are quite likely understandable, albeit not always grammatical.

	EN→UK			EN→BE		
	10K	100K	1M	10K	100K	1M
raw	6.1	13.5	15.3	2.3	8.8	9.8
lemma	12.8	19.5	21.3	3.5	13.7	15.8

Table 4: BLEU scores on raw vs lemmatized text with LANGVARMT.

Translation of Rare Words On the outputs of the EN→UK model, trained with 100K UK sentences, we compute the translation accuracy of words based on their frequency in the TGT monolingual corpus for LANGVARMT, our best baseline SUP(SRC→STD)+UNSUP(SRC→TGT) and the best performing ablation SOFTMAX. These results, shown in Table 5, reveal that LANGVARMT is more accurate at translating rare words (with frequency less than 10) compared to the baselines.

Examples We provide some examples of EN-UK and EN-Beirut Arabic translations generated by the three models in Tables 6 and 7. As evaluated by native speakers of the Beirut Arabic, we find that

frequency	PIVOT	SOFTMAX	LANGVARMT
1	0.0429	0.1516	0.1812
2	0.0448	0.2292	0.2556
3	0.0597	0.2246	0.2076
4	0.0692	0.2601	0.2962
[5,10)	0.0582	0.2457	0.2722
[10,100)	0.1194	0.2881	0.2827
[100,1000)	0.2712	0.4537	0.4449

Table 5: Translation accuracies of words based on their frequencies on EN→UK with 100K UK sentences.

despite a BLEU score of only 8, in a majority of cases our baseline model is able to generate fluent translations of the input, preserving most of the content, whereas the baseline model ignores many of the content words. We also observe that in some cases, despite predicting in the right semantic space of the pretrained embeddings, it fails to predict the right token, resulting in surface form errors (e.g., predicting adjectival forms of verbs). This phenomenon is known and studied in more detail in Kumar and Tsvetkov (2019).

E Negative Results

We present results for the following experiments: (a) adapting an English to Thai (EN→TH) model to Lao (LO). We use a parallel corpus of around 10M sentences for training the supervised EN→TH model from the CCAligned corpus (El-Kishky et al., 2020), around 140K LO monolingual sentences from the OSCAR corpus (Ortiz Suárez et al., 2020) and TED2020 dev/tests for both TH and LO[6] (Reimers and Gurevych, 2020). (b) adapting an English to Amharic Model (EN→AM) to Tigrinya (TI). We use training, development and test sets from the JW300 corpus (Agić and Vulić, 2019) containing 500K EN–AM parallel corpus and 100K Tigrinya monolingual sentences.

As summarized in Table 8, our method fails to perform well on these sets of languages. Although Thai and Lao are very closely related languages, we attribute this result to little subword overlap in their respective vocabularies which degrade the quality of the embeddings. This is because Lao's writing system is developed phonetically whereas Thai writing contains many silent characters. Considering shared phonetic information while learning the embeddings can alleviate this issue and is an avenue for future work. On the other hand, Amharic and Tigrinya, while sharing a decent amount of vocabulary, use different constructs and function words (Kidane et al., 2021) leading to a very noisy psuedo-parallel corpus.

[6]Although Thai and Lao scripts look very similar, they use different Unicode symbols which are one-to-one mappable to each other: https://en.wikipedia.org/wiki/Lao_(Unicode_block)

Source	And we never think about the hidden connection
Reference	Та ми ніколи не думаємо про приховані зв'язки
PIVOT	І ми ніколи не дуємо про приховану зв'язку. (And we never think about a hidden connection.)
SOFTMAX	Я ніколи не думав про прихований зв'язок. (I never thought of a hidden connection.)
LANGVARMT	І ми ніколи не думаємо про прихований зв'язок. (And we never think about a hidden connection.)
Source	And yet, looking at them, you would see a machine and a molecule.
Reference	Дивлячись на них, ви побачите машину і молекулу.
PIVOT	І бачити, дивлячись на них, ви бачите машину і молекулу молекули. (And to see, looking at them, you see a machine and a molecule of a molecule.)
SOFTMAX	І так, дивлячись на них, ви бачите машину і молекулу. (And so, looking at them, you see a machine and a molecule.)
LANGVARMT	І дивляючись на них, ви побачите машину і молекулу. (And looking at them, you will see a machine and a molecule)
Source	They have exactly the same amount of carbon.
Reference	Вони мають однакову кількість вуглецю.
PIVOT	Таким чином, їх частка вуглецю. (Thus, their share of carbon.)
SOFTMAX	Вони мають однакову кількість вуглецю. (They have the same amount of carbon.)
LANGVARMT	Вони мають точно таку ж кількість вуглецю. (they have exactly the same amount of carbon)

Table 6: Examples of EN-UK translations generated by LANGVARMT and the best performing baselines.

F Measuring Unfairness

When evaluating multilingual and multi-dialect systems, it is crucial that the evaluation takes into account principles of fairness, as outlined in economics and social choice theory (Choudhury and Deshpande, 2021). We follow the least difference

Source	I've never heard of this address near here.
Reference	هاي في ناونعلاهب تعمس طق ام لبق نم ةقطنملا.
PIVOT	كملسيد حر. (He will hand over.)
SOFTMAX	ناونعلاه نع تعمس قرم لاو هني. (Not once did I hear this title here) ناونعه نم أدبأ تعمس ام نوه نم بيرق. (I've never heard from this address near here.)
Source	What's the exchange rate today?
Reference	مويلا رعسلا ونش؟
PIVOT	مويلا رعس؟ (What's the rate?)
SOFTMAX	مويلا فرصلا رعس ونش؟ (What's the exchange rate today?) مويلا فرصلا رعس وش؟ (What's the exchange rate today?)
Source	How do I get to that place?
Reference	حرطملاهلا لصوب فيك؟
PIVOT	حصنتب فيك؟ (How do you recommend?)
SOFTMAX	لحملاك لصوأ فيف فيك؟ (How can I get to the shop?) لصو فيف فيك؟ (How can I get there?)
Source	Tell me when we get to the museum.
Reference	فحتملاك لصوذ سب يلق .
PIVOT	يناتلاك حورذ حر (we will go to the other.)
SOFTMAX	فحتملاك لصوذ ىتميا يكحا. (Talk when we get to the museum) فحتملا انلصو ىتميا يلق. (Tell me when we got to the museum)
Source	Please take me to the morning market.
Reference	حبصلا قوسلا ىلا يندخ فورعم لومع يرنطذ حر. (We'll wait)
SOFTMAX	حبصلا قوسلاك يندخاتنم. (You take us to the market this morning.) حبصلا قوسلاك يندخاتل لضفنم. (We prefer you take us to the market at the morning.)

Table 7: Examples of English to Beirut Arabic translations generated by LANGVARMT and the best performing baselines.

	EN→LO	EN→TI
SRC→STD	0.7	1.8
SOFTMAX	1.4	2.9
LANGVARMT	4.5	3.8

Table 8: BLEU scores for English to Lao and English to Tigrinya translation

principle proposed by Rawls (1999), whose egalitarian approach proposes to narrow the gap between unequal accuracies.

A simple proxy for unfairness is the standard deviation (or, even simpler, a $\max - \min$ performance) of the scores across languages. Beyond that, we measure a system's *unfairness* with respect to the different subgroups using the adaptation of generalized entropy index described by Speicher et al. (2018), which considers equities within and between subgroups in evaluating the overall unfairness of an algorithm on a population. The generalized entropy index for a population of n individuals receiving benefits $b_1, b_2, \ldots, b_n$ with mean benefit μ is

$$\mathcal{E}^{\alpha}(b_1, \ldots, b_n) = \frac{1}{n\alpha(\alpha-1)} \sum_{i=1}^{n} \left[\left(\frac{b_i}{\mu} \right)^{\alpha} - 1 \right].$$

Using $\alpha = 2$ following Speicher et al. (2018), the generalized entropy index corresponds to half the squared coefficient of variation.[7]

If the underlying population can be split into $|G|$ disjoint subgroups across some attribute (e.g. gender, age, or language variety) we can decompose the total unfairness into individual and group-level unfairness. Each subgroup $g \in G$ will correspond to n_g individuals with corresponding benefit vector $\mathbf{b}^g = (b_1^g, b_2^g, \ldots, b_{n_g}^g)$ and mean benefit μ_g. Then, total generalized entropy can be re-written as:

$$\begin{aligned} \mathcal{E}^{\alpha}(b_1, \ldots, b_n) =& \sum_{g=1}^{|G|} \frac{n_g}{n} \left(\frac{\mu_g}{\mu} \right)^{\alpha} \mathcal{E}^{\alpha}(\mathbf{b}^g) \\ &+ \sum_{g=1}^{|G|} \frac{n_g}{n\alpha(\alpha-1)} \left[\left(\frac{\mu_g}{\mu} \right)^{\alpha} - 1 \right] \\ =& \mathcal{E}^{\alpha}(\mathbf{b}) + \mathcal{E}_{\beta}^{\alpha}(\mathbf{b}). \end{aligned}$$

The first term $\mathcal{E}^{\alpha}(\mathbf{b})$ corresponds to the weighted unfairness score that is observed *within* each subgroup, while the second term $\mathcal{E}_{\beta}^{\alpha}(\mathbf{b})$ corresponds to the unfairness score *across* different subgroups.

In this measure of unfairness, we define the benefit as being directly proportional to the system's accuracy. For a Machine Translation system, each user receives an average benefit equal to the BLEU score the MT system achieves on the user's dialect. Conceptually, if the system produces a perfect translation (BLEU=1) then the user will receive the highest benefit of 1. If the system fails to produce a meaningful translation (BLEU$\rightarrow$ 0) then the user receives no benefit ($b = 0$) from the interaction with the system.

[7]The coefficient of variation is simply the ratio of the standard deviation σ to the mean μ of a distribution.

Is Sparse Attention more Interpretable?

Clara Meister[ϟ] **Stefan Lazov**[★] **Isabelle Augenstein**[ↀ] **Ryan Cotterell**[★,ϟ]
[ϟ]ETH Zürich [★]University of Cambridge [ↀ]University of Copenhagen
meistecl@inf.ethz.ch, stefan.lazov@cantab.net,
augenstein@di.ku.dk, ryan.cotterell@inf.ethz.ch

Abstract

Sparse attention has been claimed to increase model interpretability under the assumption that it highlights influential inputs. Yet the attention distribution is typically over representations internal to the model rather than the inputs themselves, suggesting this assumption may not have merit. We build on the recent work exploring the interpretability of attention; we design a set of experiments to help us understand how sparsity affects our ability to use attention as an explainability tool. On three text classification tasks, we verify that only a weak relationship between inputs and co-indexed intermediate representations exists—under sparse attention and otherwise. Further, we do not find any plausible mappings from sparse attention distributions to a sparse set of influential inputs through other avenues. Rather, we observe in this setting that inducing sparsity may make it less plausible that attention can be used as a tool for understanding model behavior.

1 Introduction

Interpretability research in natural language processing (NLP) is becoming increasingly important as complex models are applied to more and more downstream decision making tasks. In light of this, many researchers have turned to the attention mechanism, which has not only led to impressive performance improvements in neural models, but has also been claimed to offer insights into how models make decisions. Specifically, a number of works imply or directly state that one may inspect the attention distribution to determine the amount of influence each input token has in a model's decision-making process (Xie et al., 2017; Mullenbach et al., 2018; Niculae et al., 2018, *inter alia*).

Many lines of work have gone on to exploit this assumption when building their own "interpretable" models or analysis tools (Yang et al., 2016; Tu et al., 2016; De-Arteaga et al., 2019); one subset has even tried to make models with attention *more* interpretable by inducing sparsity—a common attribute of interpretable models (Lipton, 2018; Rudin, 2019)—in attention weights, with the motivation that this allows model decisions to be mapped to a *limited* number of items (Martins and Astudillo, 2016; Malaviya et al., 2018; Zhang et al., 2019). Yet, there lacks concrete reasoning or evidence that sparse attention weights leads to more interpretable models: customarily, attention is not directly over the model's inputs, but rather over some representation *internal* to the model, e.g. the hidden states of a recurrent network or contextual embeddings of a Transformer (see Fig. 1). Importantly, these internal representations do not solely encode information from the input token they are co-indexed with (Salehinejad et al., 2017; Brunner et al., 2020), but rather from a range of inputs. This presents the question: if internal representations themselves may not be interpretable, can we actually deduce anything from "interpretable" attention weights?

We build on the recent line of work challenging the validity of attention-as-explanation methods (Jain and Wallace, 2019; Serrano and Smith, 2019; Grimsley et al., 2020, *inter alia*) and specifically examine how *sparsity* affects their observations. To this end, we introduce a novel entropy-based metric to measure the *dispersion* of inputs' influence, rather than just their magnitudes. Through experiments on three text classification tasks, utilizing both LSTM and Transformer-based models, we observe how sparse attention affects the results of Jain and Wallace (2019) and Wiegreffe and Pinter (2019), additionally exploring whether it allows us to identify a core set of inputs that are important to models' decisions. We find we are unable to identify such a set when using sparse attention; rather, it appears that encouraging sparsity may simultaneously encourage a higher degree of

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 122–129
August 1–6, 2021. ©2021 Association for Computational Linguistics

contextualization in intermediate representations. We further observe a decrease in the correlation between the attention distribution and input feature importance measures, which exacerbates issues found by prior works. The primary conclusion of our work is that we should not believe sparse attention enhances model interpretability until we have concrete reasons to believe so; in this preliminary analysis, we do not find any such evidence.

2 Attention-based Neural Networks

We consider inputs $\mathbf{x} = x_1 \cdots x_n \in \mathcal{V}^n$ of length n where the tokens from taken from an alphabet $\mathcal{V}$. We denote the embedding of $\mathbf{x}$, e.g., its one hot encoding or (more commonly) a linear transformation of its one-hot encoding with an embedding matrix $E \in \mathbb{R}^{d \times |\mathcal{V}|}$, as $X^{(e)} \in \mathbb{R}^{d \times n}$. Our embedded input $X^{(e)}$ is then fed to an encoder, which produces n intermediate representations $I = [\mathbf{h}_1; \ldots; \mathbf{h}_n] \in \mathbb{R}^{m \times n}$, where $\mathbf{h}_i \in \mathbb{R}^m$ and m is a hyperparameter of the encoder. This transformation is quite architecture dependent.

An alignment function $A(\cdot, \cdot)$ maps a **query** $\mathbf{q}$ and a **key** K to weights $\mathbf{a}^{(t)}$ for a decoding time step t; we subsequently drop t for simplicity. In colloquial terms, A chooses which values of K should receive the most attention based on $\mathbf{q}$, which is then represented in the vector $\mathbf{a}^{(t)} \in \mathbb{R}^n$. For the NLP tasks we consider, we have $K = I = [\mathbf{h}_1; \ldots; \mathbf{h}_n]$, the encoder outputs. A query $\mathbf{q}$ may be, e.g., a representation of the question in question answering.

The weights $\mathbf{a}$ are projected to sum to 1, which results in the **attention distribution** $\boldsymbol{\alpha}$. Mathematically, this is done via a projection onto the probability simplex using a projection function ϕ, e.g., softmax or sparsemax. We then compute the **context vector** as $\mathbf{c} = \sum_{i=1}^{n} \alpha_i \, \mathbf{h}_i$. This context vector is fed to a decoder, whose structure is again architecture dependent, which generates a (possibly unnormalized) probability distribution over the set of labels $\mathcal{Y}$, where $\mathcal{Y}$ is defined by the task.

Attention. We experiment with two methods of constructing an attention distribution: (1) additive attention, proposed by Bahdanau et al. (2015): $A(K, \mathbf{q})_i = \mathbf{v}^\top \tanh(W_1 K_i + W_2 \mathbf{q})$ and (2) the scaled dot product alignment function, as in the Transformer network: $A(K, \mathbf{q}) = \frac{K^\top \mathbf{q}}{\sqrt{m}}$ where $\mathbf{v} \in \mathbb{R}^l$ and $W_1, W_2 \in \mathbb{R}^{l \times m}$ are weight matrices. Note that the original (without attention) neural encoder–decoder architecture, as in Sutskever et al. (2014), can be recovered with alignment function $A(\cdot, \cdot) = [0, \ldots, 0, 1]$, i.e., only the last of the n intermediate representations is given to the decoder.

Projection Functions. A projection function ϕ takes the output of the alignment function and maps it to a valid probability distribution: $\phi : \mathbb{R}^n \to \Delta^{n-1}$. The standard projection function is softmax:

$$\phi_{\text{soft}}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\sum_{i \in [n]} \exp(z_i)} \tag{1}$$
$$= \operatorname*{argmin}_{\mathbf{p} \in \Delta^{n-1}} \left(\sum_{i \in [n]} p_i \log p_i - \mathbf{p}^\top \mathbf{z} \right)$$

However, softmax leads to non-sparse solutions as an entry $\phi_{\text{soft}}(\mathbf{z})_i$ can only be 0 if $x_i = -\infty$. Alternatively, Martins and Astudillo (2016) introduce **sparsemax**, which can output sparse distributions:

$$\phi_{\text{sparse}}(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{p} \in \Delta^{n-1}} \|\mathbf{p} - \mathbf{z}\|_2^2 \tag{2}$$

In words, sparsemax directly maps $\mathbf{z}$ onto the probability simplex, which often leads to solutions on the boundary, i.e. where at least one entry of $\mathbf{p}$ is 0. One shortcoming of sparsemax is the lack of control over the degree of sparsity. **Sparsegen** (Laha et al., 2018) fills this gap:

$$\phi_{\text{sparseg}}(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{p} \in \Delta^{n-1}} \|\mathbf{p} - g(\mathbf{z})\|_2^2 - \lambda \|\mathbf{p}\|_2^2 \tag{3}$$

where the degree of sparsity can be tuned via the hyperparameter $\lambda \in (-\infty, 1)$; a larger λ encourages more sparsity in the minimizing solution.

3 Model Interpretability

Model interpretability and explainability have been framed in different ways (Gehrmann et al., 2019)—as model understanding tasks, where (spurious) features learned by a model are identified, or as decision understanding tasks, where explanations for particular instances are produced. We consider the latter in this paper. Such tasks can be framed as generative, where models generate free text explanations (Camburu et al., 2018; Kotonya and Toni, 2020; Atanasova et al., 2020b), or as post-hoc interpretability methods, where salient portions of the input are highlighted (Lipton, 2018; DeYoung et al., 2020; Atanasova et al., 2020a).

As there does not exist a clearly superior choice for framing decision understanding for NLP tasks (Miller, 2019; Carton et al., 2020; Jacovi and

Goldberg, 2021), we follow a substantial body of prior work in considering the post-hoc definition of interpretability based on local methods proposed by Lipton (2018). This definition is naturally operationalized through feature importance metrics and meta models (Jacovi and Goldberg, 2020). Further, we acknowledge the specific requirement that an interpretable model obeys some set of structural constraints of the domain in which it is used, such as monotonicity or physical constraints (Rudin, 2019). For NLP tasks such as sentiment analysis or topic classification, such constraints may logically include the utilization of *only* a few key words in the input when making a decision, in which case, knowing the magnitude of the influence each input token has on a model's prediction through, e.g., feature importance metrics, may suffice to verify the model obeys such constraints. While this collective definition is limited (Doshi-Velez and Kim, 2017; Guidotti et al., 2018; Rudin, 2019), we posit that if attention cannot provide model interpretability at this level, then it would likewise not be able to under more rigorous constraints.

3.1 Measures of Feature Importance

Gradient-Based Methods. Gradient-based measures of feature importance (F1; Baehrens et al., 2010; Simonyan et al., 2014; Poerner et al., 2018) use the gradient of a function's output w.r.t. a feature to measure the importance of that feature. In the case of an attentional neural network for binary classification $f(\cdot)$, we can take the gradient of f w.r.t. the variable $\mathbf{x}$ and evaluate at a point $\mathbf{x} = \mathbf{x}'$ to gain a sense of how much influence each x'_i had on the outcome $\hat{y} = f(\mathbf{x}')$. These measures are not restricted to the relationship between inputs x_i and the outcome $f(\mathbf{x})$; they can also be adapted to measure for effects from and to intermediate representations $\mathbf{h}_p$. Formally, our measures are as follows:

$$\mathbf{g}_{\hat{y}}(x_i) = \frac{\left\| \frac{\partial f}{\partial X_i^{(e)}} \right\|_2}{\sum_{k=1}^{n} \left\| \frac{\partial f}{\partial X_k^{(e)}} \right\|_2} \tag{4}$$

$$\mathbf{g}_{\mathbf{h}_p}(x_i) = \frac{\left\| \frac{\partial ||\mathbf{h}_p||_2}{\partial X_i^{(e)}} \right\|_2}{\sum_{k=1}^{n} \left\| \frac{\partial ||\mathbf{h}_p||_2}{\partial X_k^{(e)}} \right\|_2} \tag{5}$$

where $\mathbf{g}_{\hat{y}}(x_i) \in [0, 1]$ and $\mathbf{g}_{x_i}(\mathbf{h}_p) \in [0, 1]$ represents the gradient-based FI of token x_i on $\hat{y}$ and intermediate representation $\mathbf{h}_p$, respectively. Gradient-based methods are often used in explainability techniques, as they have exhibited higher correlation with human judgement than others (Atanasova et al., 2020a). Note that we take gradients w.r.t. the embedding of token x_i and that in the latter metric, we measure the influence of x_i on the magnitude of $\mathbf{h}_p$—a decision we discuss in App. A.

Leave-One-Out (LOO)-based Methods. As a secondary FI metric, we observe how model predictions change when a specific input token is removed. For token x_i, this can be calculated as:

$$D_{\hat{y}}(x_i) = \frac{|\hat{y} - \hat{y}_{-i}|}{\sum_{k=1}^{n} |\hat{y} - \hat{y}_{-k}|} \tag{6}$$

where $\hat{y}_{-i}$ is the prediction of a model with input x_i removed. The formula can also be used for intermediate representations; we denote this as $D_{\hat{y}}(\mathbf{h}_i)$.

4 Experiments

Setup. We run experiments across several model architectures, attention mechanisms, and datasets in order to understand the effects of induced attentional sparsity on model interpretability. We use three binary classification datasets: ImDB and SST (sentiment analysis) and 20News (topic classification). We use the dataset versions provided by Jain and Wallace (2019), exactly following their pre-processing steps. We show a subset of representative results here, with additional results in App. C. Further details, including model architecture descriptions, dataset statistics and baselines accuracies may be found in App. B.

Inputs and Intermediate Representations are not Interchangeable. We first explore how strongly-related inputs are to their co-indexed intermediate representations. A strong relationship on its own may validate the use of sparse attention, as the ability to identify a subset of influential intermediate representations would then directly translate to a set of influential inputs. Previous works show that the "contribution" of a token x_i to its intermediate representation $\mathbf{h}_i$ is often quite low for various model architectures (Salehinejad et al., 2017; Ming et al., 2017; Brunner et al., 2020; Tutek and Snajder, 2020). In the context of attention, we find this property to be evinced by the adversarial experiments of Wiegreffe and Pinter (2019) (§4) and Jain and Wallace (2019) (§4), which we verify in App. C. They construct adversarial attention distributions by optimizing for divergence from a baseline

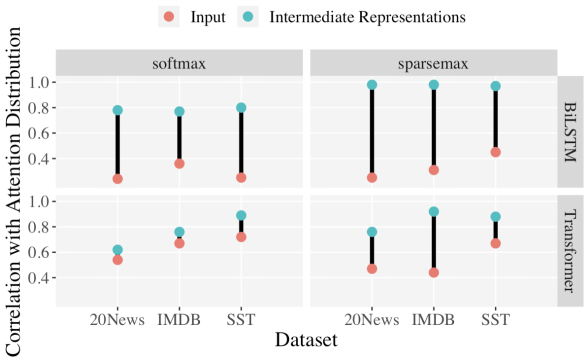


Figure 1: Correlation between the attention distribution and gradient-based FI measures. We see a notably stronger correlation between attention and FI of intermediate representation than of inputs across all models.

	IMDb	20-News	SST
	$\tilde{\mathrm{H}}(\mathbf{g}_{\mathbf{h}_i}(\mathbf{x}))$	$\tilde{\mathrm{H}}(\mathbf{g}_{\mathbf{h}_i}(\mathbf{x}))$	$\tilde{\mathrm{H}}(\mathbf{g}_{\mathbf{h}_i}(\mathbf{x}))$
BiLSTM (Softmax)	0.71 $\pm$ **0.09**	0.75 $\pm$ **0.12**	0.93 $\pm$ **0.05**
BiLSTM (Sparsemax)	0.72 $\pm$ **0.10**	0.68 $\pm$ **0.12**	0.91 $\pm$ **0.07**
Transformer (Softmax)	0.76 $\pm$ **0.08**	0.48 $\pm$ **0.06**	0.73 $\pm$ **0.09**
Transformer (Sparsemax)	0.72 $\pm$ **0.09**	0.46 $\pm$ **0.06**	0.63 $\pm$ **0.08**

Table 1: Mean entropy of gradient-based FI of input to intermediate representations. Green numbers are std. deviations. Projection functions are parenthesized.

model's attention distribution by: (1) adopting all of the baseline model's parameters and directly optimizing for divergence and (2) training an entirely new model and optimizing for divergence as part of the training process. The former method leads to a large drop in performance (accuracy) while the latter does not. If we believe the model must encode the same information to achieve similar accuracy, this discrepancy implies that in the latter method, the model likely "redistributes" information across encoder outputs (i.e., intermediate representations $\mathbf{h}_p$), which would suggest token-level information is not tied to a particular $\mathbf{h}_p$.

As further verification of high degrees of contextualization in attentional models, we report a novel quantification, offering insights into whether individual intermediate representations can be linked primarily to *any* single input—i.e., perhaps not the co-indexed input; we measure the normalized entropy[1] of the gradient-based FI of inputs to intermediate representations $\tilde{\mathrm{H}}(\mathbf{g}_{\mathbf{h}_p}(\mathbf{x})) \in [0, 1]$ to gain a sense of how dispersed influence for intermediate representation is across inputs. A value of 1 would indicate all inputs are equally influential; a value of 0 would indicate solely a single input

[1]We use Shannon entropy $\tilde{\mathrm{H}}(p) := -\sum_x p(x) \log p(x)$ albeit normalized (i.e. divided) by maximum possible entropy of the distribution to control for dimension.

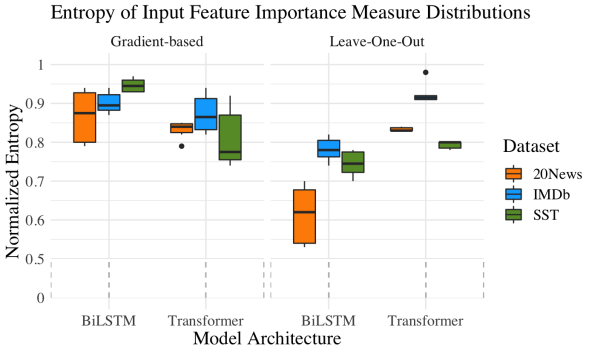


Figure 2: Entropy of gradient-based $\mathbf{g}_{\hat{y}}(\mathbf{x})$ and LOO $D_{\hat{y}}(\mathbf{x})$ FI distributions. Results are from models with full spectrum of projection functions.

	IMDb	20-News	SST
BiLSTM (tanh)	-0.935	-0.675	-0.866
Transformer (dot)	-0.830	-0.409	-0.810

Table 2: Correlation between sparsegen parameter[2] λ and entropy of gradient-based input FI $\tilde{\mathrm{H}}(\mathbf{g}_{\hat{y}}(\mathbf{x}))$.

has influence on an intermediate representation. Results in Table 1 show consistently high entropy in the distribution of the influence of inputs x_i on an intermediate representation $\mathbf{h}_p$ across all datasets, model architectures, and projection functions, which suggests the relationship between intermediate representations and inputs is far from one-to-one in these tasks.

Sparse Attention $\neq$ Sparse Input Feature Importance. Our prior results demonstrated that—even when using sparse attention—we cannot identify a subset of influential inputs directly through intermediate representations; we explore whether a subset can still be identified through FI metrics. In the case where the normalized FI distribution highlights only a few key items, the distribution will, by definition, have low entropy. Thus, we explore whether sparse attention leads to lower entropy input FI distributions in comparison to standard attention. We find no such trend; Fig. 2 shows that across all models and settings, the entropy of the FI distribution is quite high. Further, we see a consistent *negative* correlation between this entropy and the sparsity parameter of the sparsegen projection (Table 2), implying that entropy of feature importance *increases* as we raise the degree of sparsity in $\boldsymbol{\alpha}$.

Correlation between Attention and Feature Importance. Finally, we follow the experimental setup of Jain and Wallace (2019), who postulate

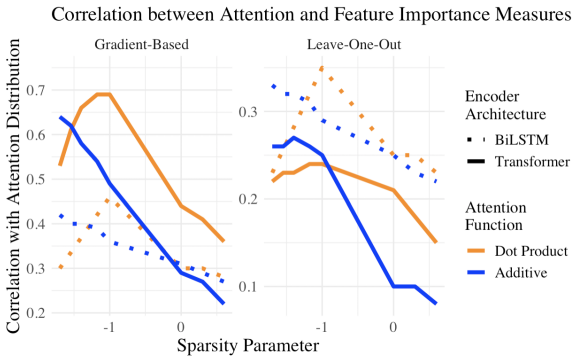


Figure 3: Correlation between the attention distribution and input FI measures as a function of the sparsity penalty λ used in the projection function ϕ_{sparseg}. x-axis is log-scaled for $\lambda < 0$ since $\lambda \in (-\infty, 1)$. Results are from the IMDb dataset.

that if the attention distribution indicates which inputs influence model behavior, then one may reasonably expect attention to correlate[2] with FI measures of the input. While they find only a weak correlation, we explore how inducing sparsity in the attention distribution affects this result. Surprisingly, Fig. 3 shows a downward trend in this correlation as the sparsity parameter λ of the sparsegen projection function is increased. As argued by Wiegreffe and Pinter (2019), a lack of this correlation does not indicate attention *cannot* be used as explanation; FI measures are not ground-truth indicators of critical inputs. However, the inverse relationship between input FI and attention is rather surprising. If anything, we may surmise sparsity in $\boldsymbol{\alpha}$ leads to *less* faithful explanations from $\boldsymbol{\alpha}$. From these results, we posit that promoting sparsity in attention distribution may simply lead to the dispersion of information to different intermediate representations, a behavior similar to that seen when constraining attention for divergence from another distribution, i.e., in the adversarial experiments of Wiegreffe and Pinter (2019) compared to those of Jain and Wallace (2019).

5 Related Work

The use of attention as an indication of inputs' influence on model decisions may at first seem natural; yet a large body of work has recently challenged this practice. Perhaps the first to do so was Jain and Wallace (2019), which revealed both a lack of correlation between the attention distribution and well established feature importance metrics and of unique optimal attention weights.[3] Subsequently, other studies arrived at similar results: Grimsley et al. (2020) found evidence that causal explanations are not attainable from attention layers over text data; Jacovi and Goldberg (2020) explored the faithfulness of attention heatmaps; Pruthi et al. (2020) showed that attention masks can be trained to give deceptive explanations. We view this work as another such study, exploring attention's innate interpretability on a different axis.

Further, this work fits into the context of a larger body of interpretability research in NLP, which has challenged the informal use of terms such as faithfulness, plausibility, and explainability (Lipton, 2018; Arrieta et al., 2020; Jacovi and Goldberg, 2021, *inter alia*) and tried to quantify the reliability of current definitions (Atanasova et al., 2020a). While we consider their findings in our experimental design—e.g., in our choice of feature importance metrics—we recognize that further experiments would be needed to address all of their concerns; for example, this work could be extended by using the benchmark created by DeYoung et al. (2020) as an additional metric of interpretability.

6 Conclusion

Prior work has cited interpretability as a driving factor for promoting sparsity in attention distributions. We explore how induced sparsity affects the ability to use attention as a tool for explaining model decisions. In our experiments on text classification tasks, we see that while sparse attention distributions may allow us to pinpoint influential intermediate representations, we are unable to find any plausible mapping from sparse attention to a small, critical set of influential inputs. Rather, we find evidence that inducing sparsity may make it even less plausible to use the attention distribution to interpret model behavior. We conclude that we need further reason to believe sparse attention increases model interpretability as our results do not support such claims.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback on the manuscript. Isabelle Augenstein's research is partially funded by a DFF Sapere Aude research leader grant.

[2] We use Kendall's τ-b correlation (Knight, 1966).

[3] Serrano and Smith (2019) contemporaneously found similar results.

Ethical Considerations

Machine learning models are being deployed in an increasing number of sensitive situations. In these settings, it is critical that models are interpretable, so that we can avoid e.g., inadvertent racial or gender bias. Giving a false sense of interpretability can allow models with undesirable (i.e., unethical or unstable) behavior to fly under the radar. We view this work as another critique of interpretability claims and hope our results will encourage the more careful consideration of interpretability assumptions when using machine learning models in practice.

References

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion*, 58.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020a. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020b. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Association for Computational Linguistics.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in Transformers. In *8th International Conference on Learning Representations*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307, Association for Computational Linguistics.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, page 120–128, Association for Computing Machinery.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *CoRR*, abs/1702.08608.

Sebastian Gehrmann, Hendrik Strobelt, Robert Krueger, Hanspeter Pfister, and Alexander M Rush. 2019. Visual interaction with deep learning models through collaborative semantic inference. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):884–894.

Christopher Grimsley, Elijah Mayfield, and Julia R.S. Bursten. 2020. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1780–1790, European Language Resources Association.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5).

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Association for Computational Linguistics.

Alon Jacovi and Yoav Goldberg. 2021. Aligning Faithful Interpretations with their Social Attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan-*

guage Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

William R. Knight. 1966. A computer method for calculating Kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, International Committee on Computational Linguistics.

Anirban Laha, Saneem Ahmed Chemmengath, Priyanka Agrawal, Mitesh Khapra, Karthik Sankaranarayanan, and Harish G Ramaswamy. 2018. On controllable sparse alternatives to softmax. In *Advances in Neural Information Processing Systems 31*, pages 6422–6432. Curran Associates, Inc.

Zachary C. Lipton. 2018. The mythos of model interpretability. *Queue*, 16(3):31–57.

Chaitanya Malaviya, Pedro Ferreira, and André F. T. Martins. 2018. Sparse and constrained attention for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 370–376, Association for Computational Linguistics.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. 2017. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology*, pages 13–24.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, Association for Computational Linguistics.

Vlad Niculae, André Martins, Mathieu Blondel, and Claire Cardie. 2018. SparseMAP: Differentiable sparse structured inference. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3799–3808.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350, Association for Computational Linguistics.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Association for Computational Linguistics.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and beyond. In *6th International Conference on Learning Representations*.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2017. Recent advances in recurrent neural networks. *CoRR*, abs/1801.01078.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Association for Computational Linguistics.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 76–85, Association for Computational Linguistics.

Martin Tutek and Jan Snajder. 2020. Staying true to your word: (how) can attention become explanation? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 131–142, Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Association for Computational Linguistics.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 950–962, Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, Association for Computational Linguistics.

J. Zhang, Y. Zhao, H. Li, and C. Zong. 2019. Attention with sparsity regularization for neural machine translation and summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(3):507–518.

The Case for Translation-Invariant Self-Attention in Transformer-Based Language Models

Ulme Wennberg **Gustav Eje Henter**
Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Sweden
{ulme, ghe}@kth.se

Abstract

Mechanisms for encoding positional information are central for transformer-based language models. In this paper, we analyze the position embeddings of existing language models, finding strong evidence of translation invariance, both for the embeddings themselves and for their effect on self-attention. The degree of translation invariance increases during training and correlates positively with model performance. Our findings lead us to propose translation-invariant self-attention (TISA), which accounts for the relative position between tokens in an interpretable fashion without needing conventional position embeddings. Our proposal has several theoretical advantages over existing position-representation approaches. Experiments show that it improves on regular ALBERT on GLUE tasks, while only adding orders of magnitude less positional parameters.

1 Introduction

The recent introduction of transformer-based language models by Vaswani et al. (2017) has set new benchmarks in language processing tasks such as machine translation (Lample et al., 2018; Gu et al., 2018; Edunov et al., 2018), question answering (Yamada et al., 2020), and information extraction (Wadden et al., 2019; Lin et al., 2020). However, because of the non-sequential and position-independent nature of the internal components of transformers, additional mechanisms are needed to enable models to take word order into account.

Liu et al. (2020) identified three important criteria for ideal position encoding: Approaches should be *inductive*, meaning that they can handle sequences and linguistic dependencies of arbitrary length, *data-driven*, meaning that positional dependencies are learned from data, and *efficient* in terms of the number of trainable parameters. Separately, Shaw et al. (2018) argued for *translation-invariant* positional dependencies that depend on the relative distances between words rather than their absolute positions in the current text fragment. It is also important that approaches be *parallelizable*, and ideally also *interpretable*. Unfortunately, none of the existing approaches for modeling positional dependencies satisfy all these criteria, as shown in Table 1 and in Sec. 2. This is true even for recent years' state-of-the-art models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and ELECTRA (Clark et al., 2020), which require many positional parameters but still cannot handle arbitrary-length sequences.

This paper makes two main contributions: First, in Sec. 3, we analyze the learned position embeddings in major transformer-based language models. Second, in Sec. 4, we leverage our findings to propose a new positional-dependence mechanism that satisfies all desiderata enumerated above. Experiments verify that this mechanism can be used alongside conventional position embeddings to improve downstream performance. Our code is available.

2 Background

Transformer-based language models (Vaswani et al., 2017) have significantly improved modeling accuracy over previous state-of-the-art models like ELMo (Peters et al., 2018). However, the non-sequential nature of transformers created a need for other mechanisms to inject positional information into the architecture. This is now an area of active research, which the rest of this section will review.

The original paper by Vaswani et al. (2017) proposed summing each token embedding with a position embedding, and then used the resulting embedding as the input into the first layer of the model. BERT (Devlin et al., 2019) reached improved performance training data-driven d-dimensional em-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 130–140
August 1–6, 2021. ©2021 Association for Computational Linguistics

Method	Induct-ive?	Data-driven?	Parameter efficient?	Translation invariant?	Parallel -izable?	Interpret-able?
Sinusoidal position embedding (Vaswani et al., 2017)	✓	✗	✓	✗	✓	✗
Absolute position embedding (Devlin et al., 2019)	✗	✓	✗	✗	✓	✗
Relative position embedding (Shaw et al., 2018)	✗	✓	✓	✓	✗	✗
T5 (Raffel et al., 2020)	✗	✓	✓	✓	✓	✓
Flow-based (Liu et al., 2020)	✓	✓	✓	✗	✗	✗
Synthesizer (Tay et al., 2020)	✗	✓	✓	✗	✓	✗
Untied positional scoring (Ke et al., 2021)	✗	✓	✗	✗	✓	✗
Rotary position embedding (Su et al., 2021)	✓	✗	✓	✓	✓	✗
Translation-invariant self-attention (proposed)	✓	✓	✓	✓	✓	✓

Table 1: Characteristics of position-representation approaches for different language-modeling architectures.

beddings for each position in text snippets of at most n tokens. A family of models have tweaked the BERT recipe to improve performance, including RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020), where the latter has layers share the same parameters to achieve a more compact model.

All these recent data-driven approaches are restricted to fixed max sequence lengths of n tokens or less (typically $n = 512$). Longformer (Beltagy et al., 2020) showed modeling improvements by increasing n to 4096, suggesting that the cap on sequence length limits performance. However, the Longformer approach also increased the number of positional parameters 8-fold, as the number of parameters scales linearly with n; cf. Table 2.

Clark et al. (2019) and Htut et al. (2019) analyzed BERT attention, finding some attention heads to be strongly biased to local context, such as the previous or the next token. Wang and Chen (2020) found that even simple concepts such as word-order and relative distance can be hard to extract from absolute position embeddings. Shaw et al. (2018) independently proposed using relative position embeddings that depend on the signed distance between words instead of their absolute position, making local attention easier to learn. They reached improved BLEU scores in machine translation, but their approach (and refinements by Huang et al. (2019)) are hard to parallelize, which is unattractive in a world driven by parallel computing. Zeng et al. (2020) used relative attention in speech synthesis, letting each query interact with separate matrix transformations for each key vector, depending on their relative-distance offset. Raffel et al. (2020) directly model position-to-position interactions, by splitting relative-distance offsets into q bins. These relative-attention approaches all facilitate processing sequences of arbitrary length, but can only resolve linguistic dependencies up to a fixed predefined maximum distance.

Tay et al. (2020) directly predicted both word and position contributions to the attention matrix without depending on token-to-token interactions. However, the approach is not inductive, as the size of the attention matrix is a fixed hyperparameter.

Liu et al. (2020) used sinusoidal functions with learnable parameters as position embeddings. They obtain compact yet flexible models, but use a neural ODE, which is computationally unappealing.

Ke et al. (2021) showed that self-attention works better if word and position embeddings are untied to reside in separate vector spaces, but their proposal is neither inductive nor parameter-efficient.

Su et al. (2021) propose rotating each embedding in the self-attention mechanism based on its absolute position, thereby inducing translational invariance, as the inner product of two vectors is conserved under rotations of the coordinate system. These rotations are, however, not learned.

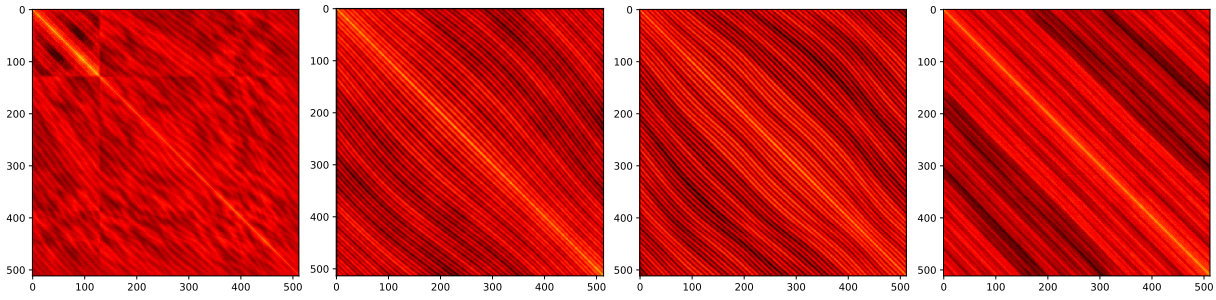
The different position-representation approaches are summarized in Table 1. None of them satisfy all design criteria. In this article, we analyze the position embeddings in transformer models, leading us to propose a new positional-scoring mechanism that combines all desirable properties (final row).

3 Analysis of Existing Language Models

In this section, we introspect selected high-profile language models to gain insight into how they have learned to account for the effect of position.

3.1 Analysis of Learned Position Embeddings

First, we stack the position embeddings in the matrix $E_P \in \mathbb{R}^{n \times d}$, and inspect the symmetric matrix $P = E_P E_P^T \in \mathbb{R}^{n \times n}$, where $P_{i,j}$ represents the inner product between the ith and jth embedding vectors. If inner products are translation invariant, $P_{i,j}$ will only depend on the difference between the indices, $j - i$, giving a *Toeplitz matrix*, a matrix where each diagonal is constant.



(a) BERT base (b) RoBERTa base (c) ALBERT base v1 (d) ALBERT xxlarge v2

Figure 1: Heatmaps visualizing the matrix $P = E_P E_P^T$ of position-embedding inner products for different models. The greater the inner product between the embeddings, the brighter the color. See appendix Figs. 4, 5 for more.

Fig. 1 visualizes the P-matrices for the position embeddings in a number of prominent transformer models, listed from oldest to newest, which also is in order of increasing performance. We note that a clear Toeplitz structure emerges from left to right. Translation invariance is also seen when plotting position-embedding cosine similarities, as done by Wang and Chen (2020) for transformer-based language models and by Dosovitskiy et al. (2020) for 2D transformers modeling image data.

In Fig. 2 we further study how the degree of Toeplitzness (quantified by R^2, the amount of the variance among matrix elements $P_{i,j}$ explained by the best-fitting Toeplitz matrix) changes for different ALBERT models. With longer training time (i.e., going from ALBERT v1 to v2), Toeplitzness increases, as the arrows show. This is associated with improved mean dev-set score. Such evolution is also observed in Wang and Chen (2020, Fig. 8).

3.2 Translation Invariance in Self-Attention

Next, we analyze how this translation invariance is reflected in self-attention. Recall that Vaswani et al. (2017) self-attention can be written as

$$\mathrm{att}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \quad (1)$$

and define position embeddings E_P, word embeddings E_W, and query and key transformation weight matrices W_Q and W_K. By taking

$$QK^T = (E_W + E_P) W_Q W_K^T (E_W + E_P)^T \quad (2)$$

and replacing each row of E_W by the average word embedding across the entire vocabulary, we obtain a matrix we call $\widehat{F}_P$ that quantifies the average effect of E_P on the softmax in Eq. (1). Plots of the resulting $\widehat{F}_P$ for all 12 ALBERT-base attention heads in the first layer are in appendix Fig. 8. Importantly, these matrices also exhibit Toeplitz structure. Fig. 3 graphs sections through the main diagonal for selected heads, showing peaks at short relative distances, echoing Clark et al. (2019) and Htut et al. (2019). In summary, we conclude that position encodings, and their effect on softmax attention, have an approximately translation-invariant structure in successful transformer-based language models.

4 Proposed Self-Attention Mechanism

We now introduce our proposal for parameterizing the positional contribution to self-attention in an efficient and translation-invariant manner, optionally removing the position embeddings entirely.

4.1 Leveraging Translation Invariance for Improved Inductive Bias

Our starting point is the derivation of Ke et al. (2021). They expand QK^T while ignoring cross terms, yielding

$$QK^T \approx E_W W_Q W_K^T E_W^T + E_P W_Q W_K^T E_P^T, \quad (3)$$

an approximation they support by theory and empirical evidence. They then "untie" the effects of words and positions by using different W-matrices for the two terms in Eq. (3). We agree with sepa-

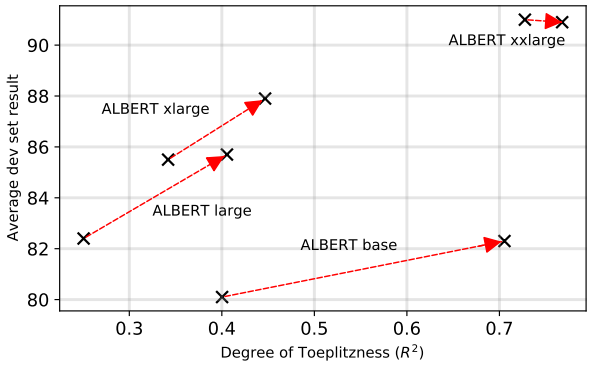


Figure 2: Scatterplot of the degree of Toeplitzness of P for different ALBERT models (v1→v2) against average performance numbers (from Lan et al.'s GitHub) over SST-2, MNLI, RACE, and SQuAD 1.1 and 2.0.

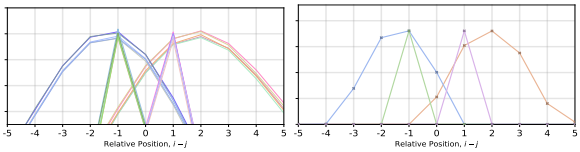


Figure 3: Positional responses of select attention heads. Left: Sections $(\widehat{F}_P)_{i,j}$ through $\widehat{F}_P$ of ALBERT base v2, varying j for 5 different i, keeping $j = i$ centered. The sections are similar regardless of i since $\widehat{F}_P$ is close to Toeplitz. Colors distinguish different heads. Right: TISA scoring functions, attending to similar positions as heads on the left. Larger plots in Figs. 6, 7.

rating these effects, but also see a chance to reduce the number of parameters.

Concretely, we propose to add a second term $F_P \in \mathbb{R}^{n\times n}$, a Toeplitz matrix, inside the parentheses of Eq. (1). F_P can either a) supplement or b) replace the effect of position embeddings on attention in our proposed model. For case a), we simply add F_P to the existing expression inside the softmax, while for case b) a term $\sqrt{d_k}F_P$ is inserted in place of the term $E_P W_Q W_K^T E_P^T$ in Eq. (3). This produces two new self-attention equations:

$$\text{att} = \begin{cases} \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + F_P\right)V & \text{a)} \\ \text{softmax}\left(\frac{Q_W K_W^T}{\sqrt{d_k}} + F_P\right)V_W & \text{b)} \end{cases} \quad (4)$$

where the inputs Q_W, K_W, and V_W (defined by $Q_W = E_W W_Q$, and similarly for K_W and V_W) do not depend on the position embeddings E_P. Case a) is not as interpretable as TISA alone (case b), since the resulting models have two terms, E_P and F_P, that share the task of modeling positional information. Our two proposals apply to any sequence model with a self-attention that follows Eq. (1), where the criteria in Table 1 are desirable.

4.2 Positional Scoring Function

Next, we propose to parameterize the Toeplitz matrix F_P using a *positional scoring function* $f_\theta(\cdot)$ on the integers $\mathbb{Z}$, such that $(F_P)_{i,j} = f_\theta(j-i)$. f_θ defines F_P-matrices of any size n. The value of $f_\theta(j-i)$ directly models the positional contribution for how the token at position i attends to position j. We call this *translation-invariant self-attention*, or TISA. TISA is inductive and can be simplified down to arbitrarily few trainable parameters.

Let $k = j - i$. Based on our findings for $\widehat{F}_P$ in Sec. 3, we seek a parametric family $\{f_\theta\}$ that allows both localized and global attention, without diverging as $|k| \to \infty$. We here study one family

	Standard	Ke et al. (2021)	TISA
General formula	nd	$nd + 2d^2$	$3SHL$
Longformer	3,145,728	4,325,376	**2,160**
BERT/RoBERTa	393,216	1,572,864	**2,160**
ALBERT	65,536	98,304	**2,160**

Table 2: Number of positional parameters for base models of different language-model architectures and different positional information processing methods, with max sequence length $n \in (512, 4096)$, position embeddings of dimension $d \in (128, 768)$, $S = 5$ kernels, $H = 12$ attention heads, and $L = 12$ layers with distinct TISA positional scoring functions. Parameter sharing gives ALBERT lower numbers. TISA can be used alone or added to the counts in other columns.

that satisfies the criteria: the radial-basis functions

$$f_\theta(k) = \sum_{s=1}^{S} a_s \exp\left(-|b_s|(k - c_s)^2\right). \quad (5)$$

Their trainable parameters are $\theta = \{a_s, b_s, c_s\}_{s=1}^S$, i.e., 3 trainable parameters per kernel s. Since these kernels are continuous functions (in contrast to the discrete bins of Raffel et al. (2020)), predictions change smoothly with distance, which seems intuitively meaningful for good generalization.

Lin et al. (2019) found that word-order information in BERTs position embeddings gets increasingly washed out from layer 4 onward. As suggested by Dehghani et al. (2019) and Lan et al. (2020), we inject positional information into each of the H heads at all L layers, resulting in one learned function $f_{\theta^{(h,l)}}$ for each head and layer. The total number of positional parameters of TISA is then $3SHL$. As seen in Table 2, this is several orders of magnitude less than the embeddings in prominent language models.

The inductivity and localized nature of TISA suggests the possibility to rapidly pre-train models on shorter text excerpts (small n), scaling up to longer n later in training and/or at application time, similar to the two-stage training scheme used by Devlin et al. (2019), but without risking the undertraining artifacts visible for BERT at $n > 128$ in Figs. 1 and 4. However, we have not conducted any experiments on the performance of this option.

5 Experiments

The main goal of our experiments is to illustrate that TISA can be added to models to improve their performance (Table 3a), while adding a minuscule amount of extra parameters. We also investigate the performance of models without position em-

Task	Baseline	$S=1$	3	5	Δ	$\Delta\%$
SST-2	92.9	**93.3**	93.1	93.1	0.4	6.5%
MNLI	83.8	84.1	84.4	**84.8**	1.0	5.9%
QQP	88.2	88.0	**88.3**	**88.3**	0.1	1.2%
STS-B	90.3	**90.4**	90.0	**90.4**	0.1	1.5%
CoLA	57.2	57.0	56.5	**58.5**	1.3	2.9%
MRPC	89.6	**90.1**	89.0	**90.1**	0.5	5.3%
QNLI	91.6	**91.7**	91.4	91.6	0.1	0.4%
RTE	72.9	71.1	**73.6**	**73.6**	0.7	2.7%

(a) ALBERT base v2 models with position embeddings

Task	Baseline	$S=1$	3	5	Δ	$\Delta\%$
SST-2	85.1	85.9	85.8	**86.0**	0.9	6.2%
MNLI	78.8	80.9	81.4	**81.6**	2.8	13.4%
QQP	86.3	86.2	86.5	**86.8**	0.5	3.4%
STS-B	89.0	89.0	**89.1**	**89.1**	0.1	0.3%
MRPC	82.8	83.1	**83.3**	83.1	0.5	3.3%
QNLI	86.6	87.2	87.4	**87.7**	1.1	7.8%
RTE	62.1	61.7	62.5	**62.8**	0.7	1.9%

(b) ALBERT base v2 models without position embeddings

Table 3: GLUE task dev-set performance (median over 5 runs) with TISA (S kernels) and without (baseline). Δ is the maximum performance increase in a row and $\Delta\%$ is the corresponding relative error reduction rate.

beddings (Table 3b), comparing TISA to a bag-of-words baseline ($S = 0$). All experiments use pretrained ALBERT base v2 implemented in Huggingface (Wolf et al., 2020). Kernel parameters $\theta^{(h)}$ for the functions in Eq. (5) were initialized by regression to the $\widehat{F}_P$ profiles of the pretrained model, (see Appendix C for details); example plots of resulting scoring functions are provided in Fig. 3. We then benchmark each configuration with and without TISA for 5 runs on GLUE tasks (Wang et al., 2018), using jiant (Phang et al., 2020) and standard dataset splits to evaluate performance.

Our results in Table 3a show relative error reductions between 0.4 and 6.5% when combining TISA and conventional position embeddings. These gains are relatively stable regardless of S. We also note that Lan et al. (2020) report 92.9 on SST-2 and 84.6 on MNLI, meaning that our contribution leads to between 1.3 and 2.8% relative error reductions over their scores. The best performing architecture ($S=5$), gives improvements over the baseline on 7 of the 8 tasks considered and on average increases the median F1 score by 0.4 points. All these gains have been realized using a very small number of added parameters, and without pre-training on any data after adding TISA to the architecture. The only joint training happens on the training data of each particular GLUE task.

Results for TISA alone, in Table 3b, are not as strong. This could be because these models are derived from an ALBERT model pretrained using conventional position embeddings, since we did not have the computational resources to tune from-scratch pretraining of TISA-only language models.

Figs. 3 and 6 plot scoring functions of different attention heads from the initialization described in Appendix C. Similar patterns arose consistently and rapidly in preliminary experiments on pretraining TISA-only models from scratch. The plots show heads specializing in different linguistic aspects, such as the previous or next token, or multiple tokens to either side, with other heads showing little or no positional dependence. This mirrors the visualizations of ALBERT base attention heads in Figs. 3, 6, 7, 8 and the findings of Htut et al. (2019) and Clark et al. (2019) on BERT, but TISA makes this directly visible in an interpretable model, without having to probe correlations in a black box.

Interestingly, the ALBERT baseline on STS-B in Table 3a is only 1.3 points ahead of the bag-of-words baseline in Table 3b. This agrees with experiments shuffling the order of words (Pham et al., 2020; Sinha et al., 2021) finding that modern language models tend to focus mainly on higher-order word co-occurrences, rather than word order, and suggests that word-order information is underutilized in state-of-the-art language models.

6 Conclusion

We have analyzed state-of-the-art transformer-based language models, finding that translation-invariant behavior emerges during training. Based on this we proposed TISA, the first positional information processing method to simultaneously satisfy the six key design criteria in Table 1. Experiments demonstrate competitive downstream performance. The method is applicable also to transformer models outside language modeling, such as modeling time series in speech or motion synthesis, or to describe dependencies between pixels in computer vision.

Acknowledgments

We would like to thank Gabriel Skantze, Dmytro Kalpakchi, Viktor Karlsson, Filip Cornell, Oliver Åstrand, and the anonymous reviewers for their constructive feedback. This research was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proc. BlackboxNLP@ACL*, pages 276–286.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *Proc. EMNLP*, pages 285–294.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers. In *Proc. ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, pages 4171–4186.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proc. EMNLP*, pages 489–500.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor O. K. Li. 2018. Meta-learning for low-resource neural machine translation. In *Proc. EMNLP*, pages 3622–3631.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in BERT track syntactic dependencies?

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music transformer. In *Proc. ICLR*.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking positional encoding in language pre-training. In *Proc. ICLR*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proc. EMNLP*, pages 5039–5049.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proc. ICLR*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proc. ACL*, pages 7999–8009.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proc. BlackboxNLP@ACL*, pages 241–253.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2020. Learning to encode position for transformer with continuous dynamical model. In *Proc. ICML*, pages 6327–6335.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. NAACL*, pages 2227–2237.

Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?

Jason Phang, Phil Yeres, Jesse Swanson, Haokun Liu, Ian F. Tenney, Phu Mon Htut, Clara Vania, Alex Wang, and Samuel R. Bowman. 2020. `jiant` 2.0: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proc. NAACL-HLT*, pages 464–468.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2020. Synthesizer: Rethinking self-attention in transformer models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NIPS*, pages 5998–6008.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proc. EMNLP-IJCNLP*, pages 5784–5789.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. BlackboxNLP@EMNLP*, pages 353–355.

Yu-An Wang and Yun-Nung Chen. 2020. What do position embeddings learn? An empirical study of pre-trained language model positional encoding. In *Proc. EMNLP*, pages 6840–6849.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proc. EMNLP System Demonstrations*, pages 38–45.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proc. EMNLP*, pages 6442–6454.

Zhen Zeng, Jianzong Wang, Ning Cheng, and Jing Xiao. 2020. Prosody Learning Mechanism for Speech Synthesis System Without Text Length Limit. In *Proc. Interspeech 2020*, pages 4422–4426.

A Visualizing $E_P E_P^T$ for Additional Language Models

Fig. 1 shows the inner product between different position embeddings for the models BERT base uncased, RoBERTa base, ALBERT base v1 as well as ALBERT xxlarge v2. Leveraging our analysis findings of translation invariance in the matrix of $E_P E_P^T$ in these pretrained networks, we investigate the generality of this phenomenon by visualizing the same matrix for additional existing large language models. We find that similar Toeplitz patterns emerge for all investigated networks.

B Coefficient of Determination R^2

The coefficient of determination, R^2, is a widely used concept in statistics that measures what fraction of the variance in a dependent variable that can be explained by an independent variable. Denoting the Residual Sum of Squares, RSS, and Total Sum of Squares, TSS, we have that

$$R^2 = 1 - \frac{RSS}{TSS}, \tag{6}$$

where $R^2 = 0$ means that the dependent variable is not at all explained, and $R^2 = 1$ means that the variance is fully explained by the independent variable.

Applied to a matrix, $A \in \mathbb{R}^{n \times n}$, to determine its degree of Toeplitzness, we get RSS by finding the Toeplitz matrix, $A_T \in \mathbb{R}^{n \times n}$, that minimizes the following expression:

$$RSS = \min_{A_T} \sum_{i=1}^{n} \sum_{j=1}^{n} (A - A_T)_{i,j}^2 \tag{7}$$

Furthermore, we can compute TSS as:

$$TSS = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(A_{i,j} - \left(\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \right) \right)^2 \tag{8}$$

C Extracting ALBERT positional scores

In order to extract out the positional contributions to the attention scores from ALBERT, we disentangle the positional and word-content contributions from equation (3), and remove any dependencies on the text sequence through E_W. We exchange $E_W \approx E_{\overline{W}}$, with the average word embedding over the *entire vocabulary*, which we call $E_{\overline{W}}$.

$$F_P \approx \frac{1}{\sqrt{d_k}} (E_W W_Q W_K^T E_P^T + \tag{9}$$

$$+ E_P W_Q W_K^T E_W^T + E_P W_Q W_K^T E_P^T) \tag{10}$$

$$\approx \frac{1}{\sqrt{d_k}} (E_{\overline{W}} W_Q W_K^T E_P^T + \tag{11}$$

$$+ E_P W_Q W_K^T E_{\overline{W}}^T + E_P W_Q W_K^T E_P^T) \tag{12}$$

This way, we can disentangle and extract the positional contributions from the ALBERT model.

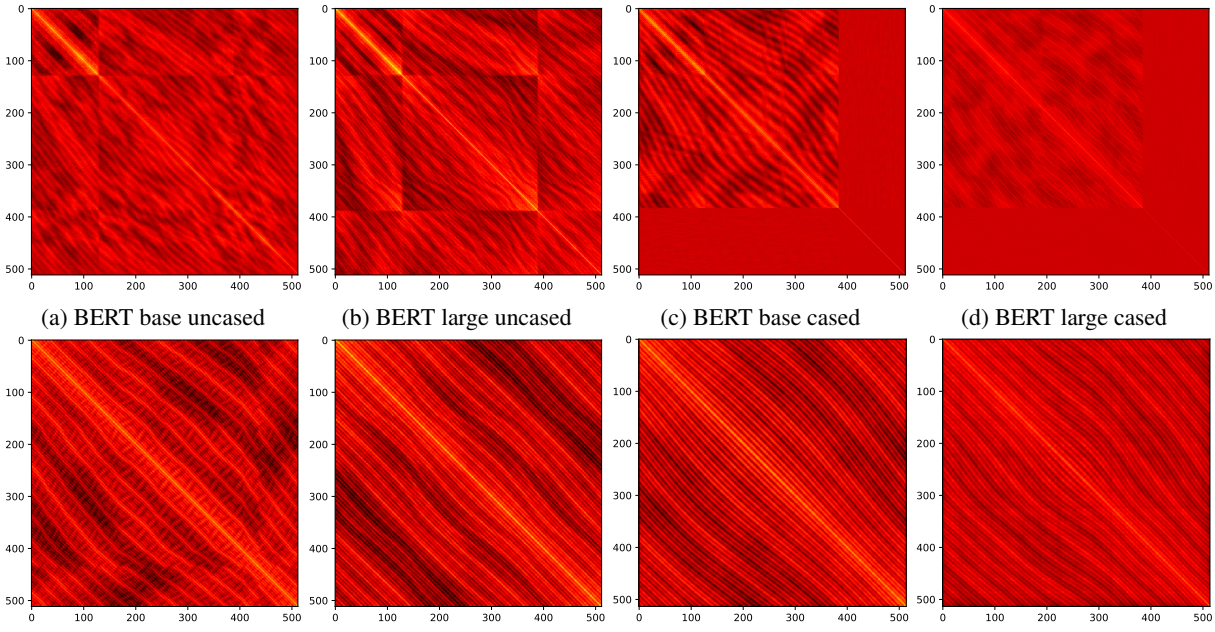
Initialization of Position-Aware Self-Attention Using this trick, we initialize F_P with formula (12). Since F_P is only generating the positional scores, which are independent of context, it allows for training a separate positional scorer neural network to predict the positional contributions in the ALBERT model. Updating only 2,160 parameters (see Table 2) significantly reduces the computational load. This pretraining initialization scheme converges in less than 20 seconds on a CPU.

Removing Position Embeddings When removing the effect of position embeddings, we calculate the average position embedding and exchange all position embeddings for it. This reduces the variation between position embeddings, while conserving the average value of the original input vectors $E_W + E_P$.

Extracted Attention Score Contributions Leveraging our analysis findings of translation invariance in large language models, we visualize the scoring functions as a function of relative distance offset between tokens. Fig. 3 shows the implied scoring functions for 4 attention heads for 5 different absolute positions. Figs. 6, 7 show all 12 attention heads of ALBERT base v2 with TISA.

D Number of Positional Parameters of Language Models

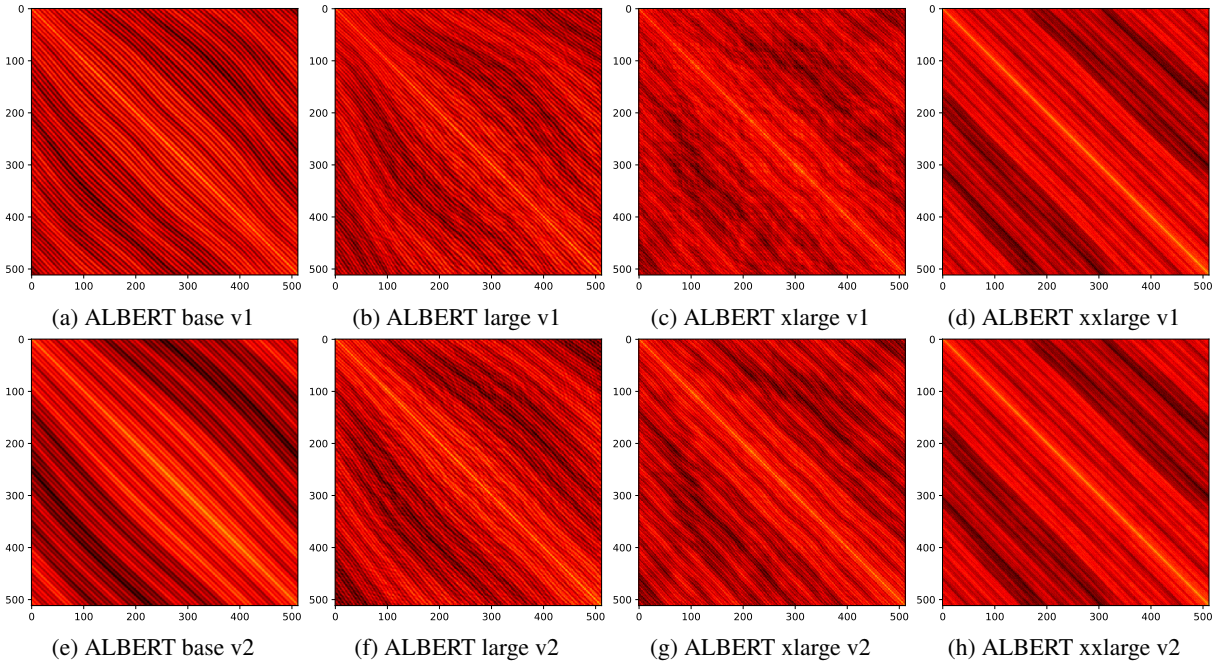
In the paper, define positional parameters as those modeling only positional dependencies. In most BERT-like models, these are the position embeddings only (typically $n \times d$ parameters). Ke et al. (2021) propose to separate position and content embeddings, yielding more expressive models with separate parts of the network for processing separate information sources. In doing so, they introduce two weight matrices specific to positional information processing, $U_Q \in \mathbb{R}^{d \times d}$ and $U_K \in \mathbb{R}^{d \times d}$, totaling $nd + 2d^2$ positional parameters.



(a) BERT base uncased (b) BERT large uncased (c) BERT base cased (d) BERT large cased

(e) ELECTRA small (f) ELECTRA large (g) RoBERTa base (h) RoBERTa large

Figure 4: Visualizations of the inner-product matrix $P = E_P E_P^T \in \mathbb{R}^{n \times n}$ for different BERT, ELECTRA, and RoBERTa models. We see that ELECTRA and RoBERTa models show much stronger signs of translational invariance than their BERT counterparts. Most BERT models follow the pattern noted by Wang and Chen (2020), where the Toeplitz structure is much more pronounced for the first 128×128 submatrix, reflecting how these models mostly were trained on 128-token sequences, and only scaled up to $n = 512$ for the last 10% of training (Devlin et al., 2019). Position embeddings 385 through 512 of the BERT cased models show a uniform color, suggesting that these embeddings are almost completely untrained.



(a) ALBERT base v1 (b) ALBERT large v1 (c) ALBERT xlarge v1 (d) ALBERT xxlarge v1

(e) ALBERT base v2 (f) ALBERT large v2 (g) ALBERT xlarge v2 (h) ALBERT xxlarge v2

Figure 5: Visualizations of the inner-product matrix $P = E_P E_P^T \in \mathbb{R}^{n \times n}$ for different ALBERT models (Lan et al., 2020). We plot both v1 and v2 to show the progression towards increased Toeplitzness during training.

Hyperparameter Selection We performed a manual hyperparameter search starting from the hyperparameters that the Lan et al. (2020) report in https://github.com/google-research/albert/blob/master/run_glue.sh. Our hyperparameter config files can be found with our code.

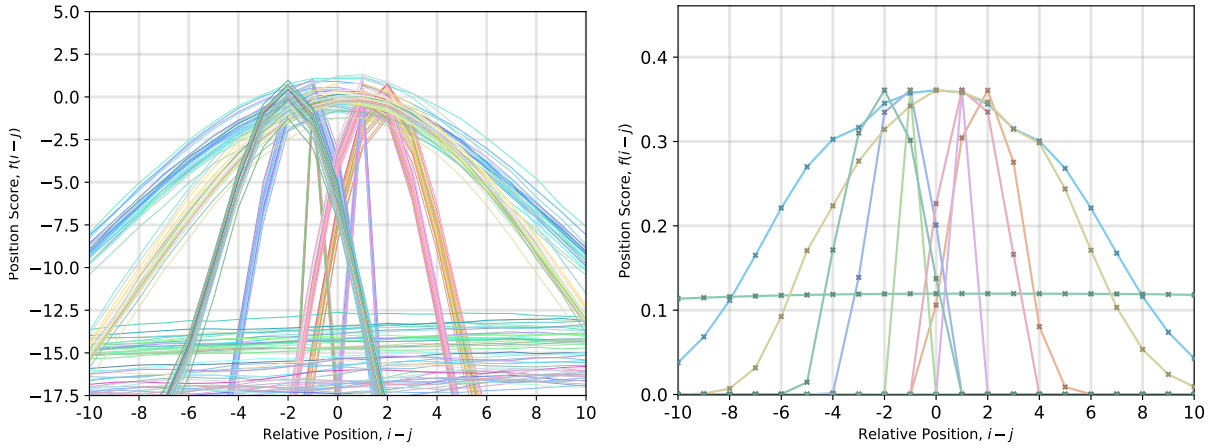
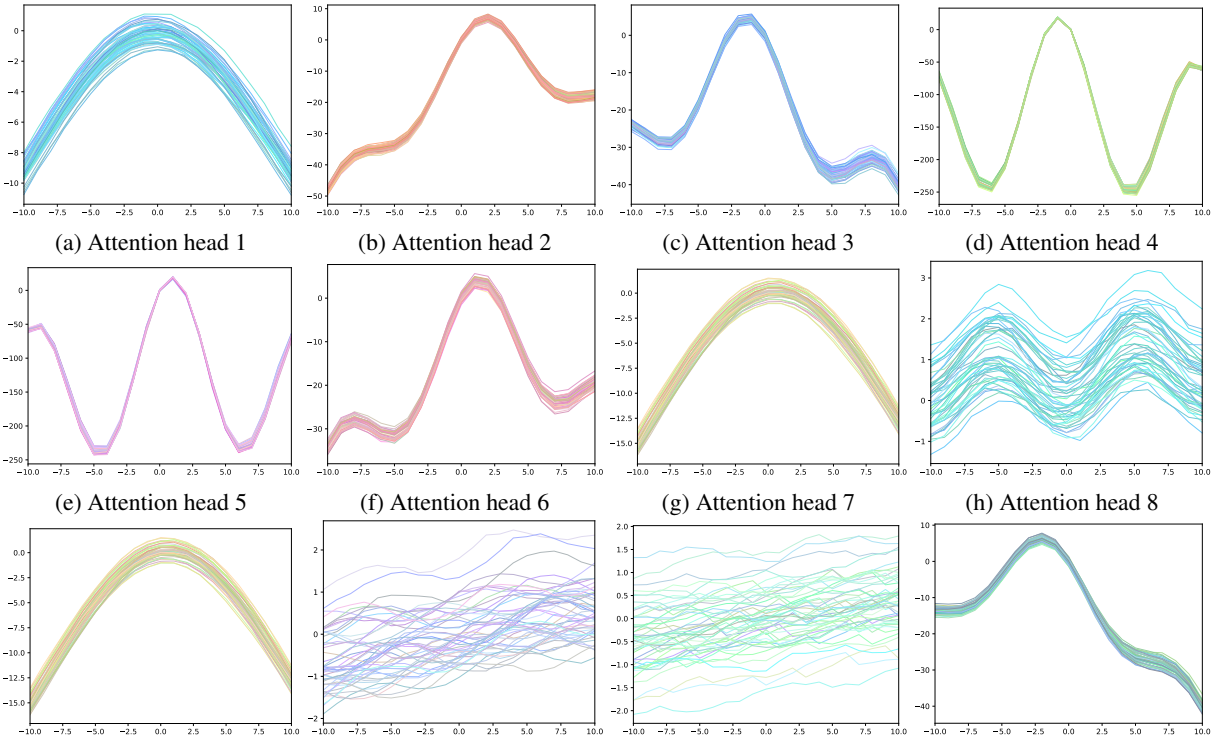


Figure 6: Positional responses of all attention heads. Sections through $\widehat{F}_P$ of ALBERT base v2, aligned to the main diagonal, (left) show similar profiles as the corresponding TISA scoring functions (right). Vertical axes differ due to 1) the scaling factor $\sqrt{d_k}$ and 2) softmax being invariant to vertical offset.



(a) Attention head 1 (b) Attention head 2 (c) Attention head 3 (d) Attention head 4

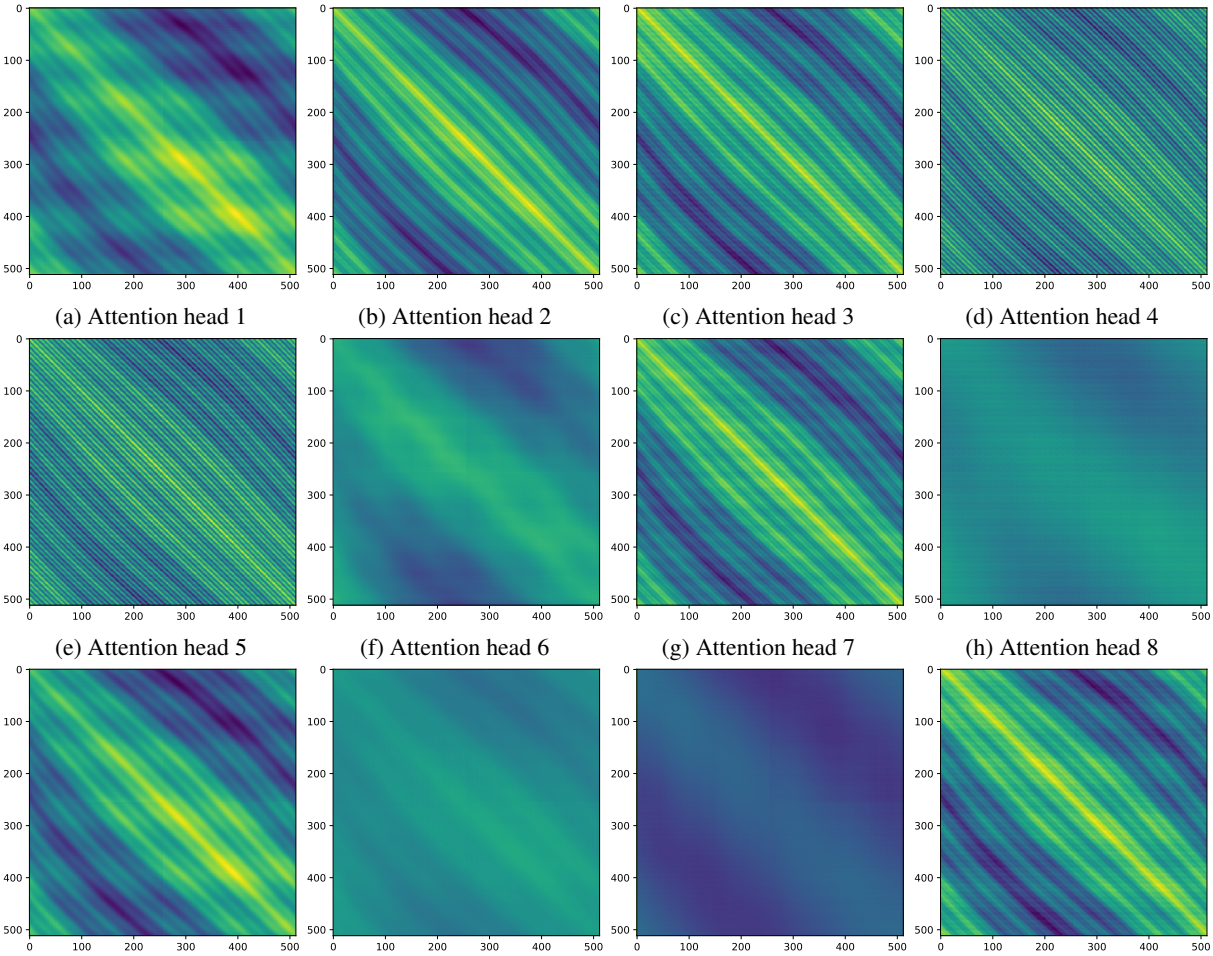
(e) Attention head 5 (f) Attention head 6 (g) Attention head 7 (h) Attention head 8

(i) Attention head 9 (j) Attention head 10 (k) Attention head 11 (l) Attention head 12

Figure 7: Rows from the positional attention matrices $\widehat{F}_P$ for all ALBERT base v2 attention heads, centered on the main diagonal. Note that the vertical scale generally differs between plots. The plots are essentially aligned sections through the matrices in Fig. 8, but zoomed in to show details over short relative distances since this is where the main peak(s) are located, and the highest values are by far the most influential on softmax attention.

E Reproducibility

Experiments were run on a GeForce RTX 2080 machine with 8 GPU-cores. Each downstream experiment took about 2 hours to run. Datasets and code can be downloaded from https://github.com/nyu-mll/jiant/blob/master/guides/tasks/supported_tasks.md and https://github.com/ulmewennberg/tisa.



(a) Attention head 1 (b) Attention head 2 (c) Attention head 3 (d) Attention head 4

(e) Attention head 5 (f) Attention head 6 (g) Attention head 7 (h) Attention head 8

(i) Attention head 9 (j) Attention head 10 (k) Attention head 11 (l) Attention head 12

Figure 8: Values extracted from the positional attention matrices for all ALBERT base v2 first-layer attention heads. Some heads are seen to be sensitive to position, while others are not. Note that these visualizations deliberately use a different color scheme from other (red) matrices, to emphasize the fact that the matrices visualized here represent a different phenomenon and are not inner products.

Relative Importance in Sentence Processing

Nora Hollenstein
Center for Language Technology
University of Copenhagen
nora.hollenstein@hum.ku.dk

Lisa Beinborn
Computational Linguistics & Text Mining Lab
Vrije Universiteit Amsterdam
l.beinborn@vu.nl

Abstract

Determining the relative importance of the elements in a sentence is a key factor for effortless natural language understanding. For human language processing, we can approximate patterns of relative importance by measuring reading fixations using eye-tracking technology. In neural language models, gradient-based saliency methods indicate the relative importance of a token for the target objective. In this work, we compare patterns of relative importance in English language processing by humans and models and analyze the underlying linguistic patterns. We find that human processing patterns in English correlate strongly with saliency-based importance in language models and not with attention-based importance. Our results indicate that saliency could be a cognitively more plausible metric for interpreting neural language models. The code is available on github: https://github.com/beinborn/relative_importance.

1 Introduction

When children learn to read, they first focus on each word individually and gradually learn to anticipate frequent patterns (Blythe and Joseph, 2011). More experienced readers are able to completely skip words that are predictable from the context and to focus on the more *relevant* words of a sentence (Schroeder et al., 2015). Psycholinguistic studies aim at unraveling the characteristics that determine the relevance of a word and find that lexical factors such as word class, word frequency, and word complexity play an important role, but that the effects vary depending on the sentential context (Rayner and Duffy, 1986).

In natural language processing, the relative importance of words is usually interpreted with respect to a specific task. Emotional adjectives are most relevant in sentiment detection (Socher et al., 2013), relative frequency of a term is an indicator for information extraction (Wu et al., 2008), the relative position of a token can be used to approximate novelty for summarisation (Chopra et al., 2016), and function words play an important role in stylistic analyses such as plagiarism detection (Stamatatos, 2011). Neural language models are trained to be a good basis for any of these tasks and are thus expected to represent a more general notion of relative importance (Devlin et al., 2019).

Relative importance of the input in neural networks can be modulated by the so-called "attention" mechanism (Bahdanau et al., 2014). Analyses of image processing models indicate that attention weights reflect cognitively plausible patterns of visual saliency (Xu et al., 2015; Coco and Keller, 2012). Recent research in language processing finds that attention weights are not a good proxy for relative importance because different attention distributions can lead to the same predictions (Jain and Wallace, 2019). Gradient-based methods such as saliency scores seem to better approximate the relative importance of input words for neural processing models (Bastings and Filippova, 2020).

In this work, we compare patterns of relative importance in human and computational English language processing. We approximate relative importance for humans as the relative fixation duration in eye-tracking data collected in naturalistic language understanding scenarios. In related work, Sood et al. (2020a) measure the correlation between attention in neural networks trained for a document-level question-answering task and find that the attention in a transformer language model deviates strongly from human fixation patterns. In this work, we instead approximate relative importance in computational models using gradient-based saliency and find that it correlates much better with human patterns.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 141–150
August 1–6, 2021. ©2021 Association for Computational Linguistics

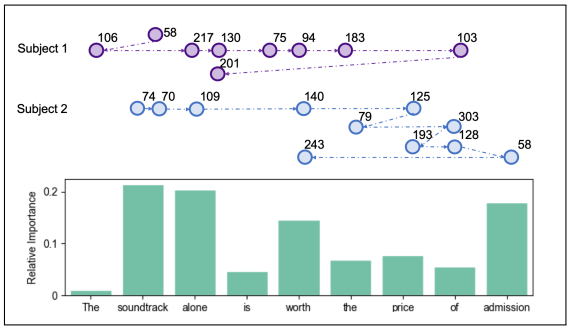


Figure 1: Example fixations for two subjects in the ZuCo dataset for the sentence "The soundtrack alone is worth the price of admission". The numbers indicate the fixation duration and the circles represent the approximate horizontal position of the fixation (positions are simplified for better visualization). The plot at the bottom indicates the relative importance of each token averaged over all subjects.

2 Determining Relative Importance

The concept of relative importance of a token for sentence processing encompasses several related psycholinguistic phenomena such as relevance for understanding the sentence, difficulty and novelty of a token within the context, semantic and syntactic surprisal, or domain-specificity of a token. We take a data-driven perspective and approximate the relative importance of a token by the processing effort that can be attributed to it compared to the other tokens in the sentence.

2.1 In Human Language Processing

The sentence processing effort can be approximated indirectly using a range of metrics such as response times in reading comprehension experiments (Su and Davison, 2019), processing duration in self-paced reading (Linzen and Jaeger, 2016), and voltage changes in electroencephalography recordings (Frank et al., 2015). In this work, we approximate relative importance using eye movement recordings during reading because they provide online measurements in a comfortable experimental setup which is more similar to a normal, uncontrolled reading experience. Eye-tracking technology can measure with high accuracy how long a reader fixates each word. The fixation duration and the relative importance of a token for the reader are strongly correlated with reading comprehension (Rayner, 1977; Malmaud et al., 2020).

Language models that look ahead and take both the left and right context into account are often considered cognitively less plausible because humans process language incrementally from left to right (Merkx and Frank, 2020). However, in human reading, we frequently find regressions: humans fixate relevant parts of the left context again while already knowing what comes next (Rayner, 1998). In Figure 1, subject 1 first reads the entire sentences and then jumps back to the token "alone". Subject 2 performs several regressions to better understand the second half of the sentence. The fixation duration is a cumulative measure that sums over these repeated fixations. Absolute fixation duration can vary strongly between subjects due to differences in reading speed but the relative fixation duration provides a good approximation for the relative importance of a token as it abstracts from individual differences. We average the relative fixation duration over all subjects to obtain a more robust signal (visualized in the plot at the bottom of Figure 1).

2.2 In Computational Language Processing

In computational language models, the interpretation of a token depends on the tokens in its context but not all tokens are equally important. To account for varying importance, so-called attention weights regulate the information flow in neural networks (Bahdanau et al., 2014). These weights are optimized with respect to a target objective and higher attention for an input token has been interpreted as higher importance with respect to the output (Vig, 2019). Recent research indicates that complementary attention distributions can lead to the same model prediction (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019) and that the removal of input tokens with large attention weights often does not lead to a change in the model's prediction (Serrano and Smith, 2019). In transformer models, the attention weights often approximate an almost uniform distribution in higher model layers (Abnar and Zuidema, 2020). Bastings and Filippova (2020) argue that saliency methods are more suitable for assigning importance weights to input tokens.

Saliency methods calculate the gradient of the output corresponding to the correct prediction with respect to an input element to identify those parts of the input that have the biggest influence on the prediction (Lipton, 2018). Saliency maps were first developed for image processing models to highlight the areas of the image that are discriminative with respect to the tested output class (Simonyan et al., 2014). Li et al. (2016) adapt this method to calculate the relative change of the output probabilities with respect to individual input tokens in

text classification tasks and Ding et al. (2019) calculate saliency maps for interpreting the alignment process in machine translation models.

In general-purpose language models such as BERT (Devlin et al., 2019), the objective function tries to predict a token based on its context. A saliency vector for a masked token thus indicates the importance of each of the tokens in the context of correctly predicting the masked token (Madsen, 2019).

We iterate over each token vector $\mathbf{x}_i$ in our input sequence $x_1, x_2, ... x_n$. Let $\mathbf{X}_i$ be the input matrix with $\mathbf{x}_i$ being masked. The saliency s_{ij} for input token $\mathbf{x}_j$ for the prediction of the correct token $\mathbf{t}_i$ is then calculated as the Euclidean norm of the gradient of the logit for x_i.

$$s_{ij} = \|\nabla_{\mathbf{x}_j} f_{t_i}(\mathbf{X}_i)\|_2 \quad (1)$$

The saliency vector $\mathbf{s}_i$ indicates the relevance of each token for the correct prediction of the masked token t_i.[1] The saliency scores are normalized by dividing by the maximum. We determine the relative importance of a token by summing over the saliency scores for each token. For comparison, we also approximate importance using attention values from the last layer of each model as Sood et al. (2020a).

2.3 Patterns of Relative Importance

Relative importance in human processing and in computational models is sensitive to linguistic properties. Rayner (1998) provides a detailed overview of token-level features that have been found to correlate with fixation duration such as length, frequency, and word class. On the contextual level, lexical and syntactic disambiguation processes cause regressions and thus lead to longer fixation duration (Just and Carpenter, 1980; Lowder et al., 2018). Computational models are also highly susceptible to frequency effects and surprisal metrics calculated using language models can predict the human processing effort (Frank et al., 2013).

The inductive bias of language processing models can be improved using the eye-tracking signal (Barrett et al., 2018; Klerke and Plank, 2019) and the modification leads to more "human-like" output in generative tasks (Takmaz et al., 2020; Sood et al., 2020b). This indicates that patterns of relative importance in computational representations differ from human processing patterns. Previous work focused on identifying links between the eye-tracking signal and attention (Sood et al., 2020a). To our knowledge, this is the first attempt to correlate fixation duration with saliency metrics.

	Dataset	BERT	Distil	ALBERT	Rand
Saliency	GECO	**.54**	.51	.48	.00
	ZuCo	**.68**	.64	.62	.00
Attention	GECO	.18	.06	.26	.00
	ZuCo	.11	.03	.37	.00

Table 1: Spearman correlation between relative fixation duration by humans and attention and saliency in the language models. Correlation values are averaged over all sentences. *Rand* is a permutation baseline.

The eye-tracking signal represents human reading processes aimed at language understanding. In previous work, we have shown that contextualized language models can predict eye patterns associated with human reading (Hollenstein et al., 2021), which indicates that computational models and humans encode similar linguistic patterns. It remains an open debate to which extent language models are able to approximate language understanding (Bender and Koller, 2020). We are convinced that language needs to be cooperatively grounded in the real world (Beinborn et al., 2018). Purely text-based language models clearly miss important aspects of language understanding but they can approximate human performance in an impressive range of processing tasks. We aim to gain a deeper understanding of the similarities and differences between human and computational language processing to better evaluate the capabilities of language models.

3 Methodology

We extract relative importance values for tokens from eye-tracking corpora and language models as described in section 2 and calculate the Spearman correlation for each sentence.[2] We first average the correlation over all sentences to analyze whether the importance patterns of humans and models are comparable and then conduct token-level analyses.

3.1 Eye-tracking Corpora

We extract the relative fixation duration from two eye-tracking corpora and average it over all readers for each sentence. Both corpora record natural reading and the text passages were followed by

[1] Our implementation adapts code from https://pypi.org/project/textualheatmap/. An alternative would be to multiply saliency and input (Alammar, 2020).

[2] Kendall's τ and KL divergence yield similar results.

multiple-choice questions to test the readers' comprehension.

GECO contains eye-tracking data from 14 native English speakers reading the entire novel *The Mysterious Affair at Styles* by Agatha Christie (Cop et al., 2017). The text was presented on the screen in paragraphs.

ZuCo contains eye-tracking data of 30 native English speakers reading full sentences from movie reviews and Wikipedia articles (Hollenstein et al., 2018, 2020).[3]

3.2 Language Models

We compare three state-of-the-art language models trained for English: BERT, ALBERT, and DistilBERT.[4] BERT was the first widely successful transformer-based language model and remains highly influential (Devlin et al., 2019). ALBERT and DistilBERT are variants of BERT that require less training time due to a considerable reduction of the training parameters while maintaining similar performance on benchmark datasets (Lan et al., 2019; Sanh et al., 2019).[5] We analyze if the lighter architectures have an influence on the patterns of relative importance that the models learn.

4 Results

The results in Table 1 show that relative fixation duration by humans strongly correlates with the saliency values of the models. In contrast, attention-based importance does not seem to be able to capture the human importance pattern. A random permutation baseline that shuffles the importance assigned by the language model yields no correlation (0.0) in all conditions.[6] As the standard deviations of the correlation across sentences are quite high (ZuCo: ∼0.22, GECO: ∼0.39), the small differences between models can be neglected (although they are consistent across corpora). For the subsequent analyses, we focus only on the BERT model which yields the best results. The differences between the corpora might be related to the number of sentences and the differences in average sentence length (ZuCo: 924, 19.5, GECO: 4,926, 12.7).

		Length		Frequency	
		Sent	Tok	Sent	Tok
GECO	Human	.69	.31	-.36	-.25
	BERT	.65	.27	-.48	-.28
ZuCO	Human	.75	.47	-.52	-.36
	BERT	.72	.36	-.65	-.40

Table 2: Spearman correlation between relative importance and word length and frequency. For the *Sent* condition, correlation is calculated per sentence and averaged. For *Tok*, importance is normalized by sentence length and correlation is calculated over all tokens.

Length and Frequency In eye-tracking data, word length correlates with fixation duration because it takes longer to read all characters. The correlation for frequency is inverse because high-frequency words (e.g. "the", "has") are often skipped in processing as they carry (almost) no meaning (Rayner, 1998). For English, word frequency and word length are both closely related to word complexity (Beinborn et al., 2014). Language models do not directly encode word length but they are sensitive to word frequency.

Our results in Table 2 show that both token length and frequency are strongly correlated with relative importance on the sentence level. Interestingly, the correlation decreases when it is calculated directly over all tokens indicating that the token-level relation between length and importance is more complex than the correlation might suggest.

Word Class Figure 2 shows the average relative importance of all tokens belonging to the same word class (normalized by sentence length). We see that both humans and BERT clearly assign higher importance to content words (left) than to function words (right). Interjections such as "Oh" in figure 3 receive the highest relevance which is understandable because they interrupt the reading flow. When we look at individual sentences, we note that the differences in importance are more pronounced in the model saliency while human fixation duration yields a smoother distribution over the tokens.

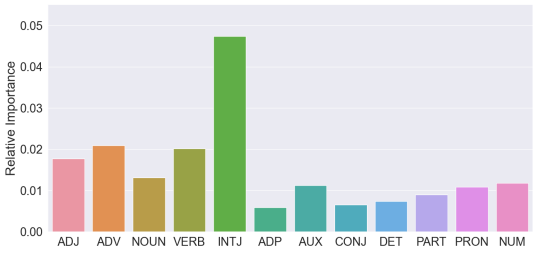
Novelty We extract the language model representations for each sentence separately whereas the readers processed the sentences consecutively. If tokens are mentioned repeatedly such as "Sherlock

[3] We combine ZuCo 1.0 (T1, T2) and ZuCo 2.0. (T1).

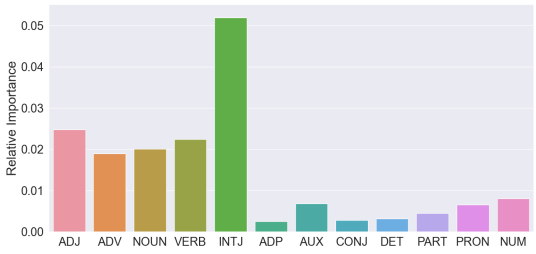
[4] We use the *Huggingface* transformers implementation (Wolf et al., 2020) and the models `bert-based-uncased`, `albert-base-v2`, and `distilbert-base-uncased`.

[5] Reduction is achieved by parameter sharing across layers (ALBERT) and by distillation which approximates the output distribution of the original BERT model using a smaller network (DistilBERT). See model references for details.

[6] We repeat the permutation 100 times and average the correlation over all iterations.



(a) Human Fixation



(b) Model saliency

Figure 2: Relative importance of tokens with respect to word class. Relative importance is measured as relative fixation duration for humans in GECO (left) and as relative gradient-based saliency in the BERT model (right).

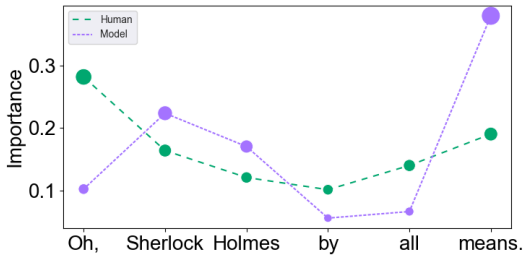


Figure 3: Relative importance values for an example sentence from the GECO corpus for the BERT model and the human values.

Holmes" which also occurred in the sentence preceding the example in Figure 3), processing ease increases for the reader, and not for the model. Some language models are able to process multiple sentences, but establishing semantic links across sentences remains a challenge.

5 Conclusion

We find that human sentence processing patterns in English correlate strongly with saliency-based importance in language models and not with attention-based importance. Our results indicate that saliency could be a cognitively more plausible metric for interpreting neural language models. In future work, it would be interesting to test the robustness of the approach with different variants for calculating saliency (Bastings and Filippova, 2020; Ding and Koehn, 2021). As we conducted our analyses only for English data, it is not yet clear whether our results generalize across languages. We will address this in future work using eye-tracking data from non-English readers (Makowski et al., 2018; Laurinavichyute et al., 2019) and comparing mono- and multilingual models (Beinborn and Choenni, 2020). We want to extend the token-level analyses to syntactic phenomena and cross-sentence effects. For example, it would be interesting to see how a language model encodes relative importance for sentences that are syntactically correct but not semantically meaningful (Gulordava et al., 2018).

Previous work has shown that the inductive bias of recurrent neural networks can be modified to obtain cognitively more plausible model decisions (Bhatt et al., 2020; Shen et al., 2019). In principle, our approach can also be applied to left-to-right models such as GPT-2 (Radford et al., 2019). In this case, the tokens at the beginning of the sentence would be assigned disproportionately high importance as the following tokens cannot contribute to the prediction of preceding tokens in incremental processing. It might thus be more useful to only use the first fixation duration of the gaze signal for analyzing importance in left-to-right models. However, we think that the regressions by the readers provide valuable information about sentence processing.

6 Ethical Considerations

Data from human participants were leveraged from freely available datasets (Hollenstein et al., 2018, 2020; Cop et al., 2017). The datasets provide anonymized records in compliance with ethical board approvals and do not contain any information that can be linked to the participants.

Acknowledgements

Lisa Beinborn's research was partially funded by the Dutch National Science Organisation (NWO) through the project CLARIAH-PLUS (CP-W6-19-005). We thank the anonymous reviewers for their constructive feedback.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Jay Alammar. 2020. Interfaces for explaining transformer language models.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312, Brussels, Belgium. Association for Computational Linguistics.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

Lisa Beinborn, Teresa Botschen, and Iryna Gurevych. 2018. Multimodal grounding for language processing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2325–2339, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lisa Beinborn and Rochelle Choenni. 2020. Semantic drift in multilingual representations. *Computational Linguistics*, 46(3):571–603.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Gantavya Bhatt, Hritik Bansal, Rishubh Singh, and Sumeet Agarwal. 2020. How much complexity does an RNN architecture need to learn syntax-sensitive dependencies? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 244–254, Online. Association for Computational Linguistics.

Hazel Blythe and Holly Joseph. 2011. *Children's Eye Movements during Reading*.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Moreno I. Coco and Frank Keller. 2012. Scan patterns predict sentence production in the cross-modal processing of visual scenes. *Cognitive Science*, 36(7):1204–1223.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior Research Methods*, 49(2):602–615.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2013. Word surprisal predicts n400 amplitude during reading. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 878–883. ACL.

Stefan L Frank, Leun J Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn. 2021. Multilingual language models predict human reading behavior. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 106–123, Online. Association for Computational Linguistics.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading. *Scientific Data*.

Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. 2020. ZuCo 2.0: A dataset of physiological recordings during natural reading and annotation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 138–146, Marseille, France. European Language Resources Association.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: From eye fixations to comprehension. *Psychological review*, 87(4):329.

Sigrid Klerke and Barbara Plank. 2019. At a glance: The impact of gaze aggregation views on syntactic tagging. In *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, pages 51–61, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

AK Laurinavichyute, Irina A Sekerina, SV Alexeeva, and KA Bagdasaryan. 2019. Russian Sentence Corpus: Benchmark measures of eye movements in reading in Cyrillic. *Behavior research methods*, 51(3):1161–1178.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Tal Linzen and T. Florian Jaeger. 2016. Uncertainty and expectation in sentence processing: Evidence from subcategorization distributions. *Cognitive Science*, 40(6):1382–1411.

Zachary C. Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Matthew W Lowder, Wonil Choi, Fernanda Ferreira, and John M Henderson. 2018. Lexical predictability during natural reading: Effects of surprisal and entropy reduction. *Cognitive science*, 42:1166–1183.

Andreas Madsen. 2019. Visualizing memorization in rnns. *Distill*. Https://distill.pub/2019/memorization-in-rnns.

Silvia Makowski, Lena A Jäger, Ahmed Abdelwahab, Niels Landwehr, and Tobias Scheffer. 2018. A discriminative model for identifying readers and assessing text comprehension from eye movements. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 209–225. Springer.

Jonathan Malmaud, Roger Levy, and Yevgeni Berzak. 2020. Bridging information-seeking human gaze and machine reading comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 142–152.

Danny Merkx and Stefan L Frank. 2020. Comparing transformers and RNNs on predicting human sentence processing data. *arXiv preprint arXiv:2005.09471*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Keith Rayner. 1977. Visual attention in reading: Eye movements reflect cognitive processes. *Memory & Cognition*, 5(4):443–448.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372.

Keith Rayner and Susan A Duffy. 1986. Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14(3):191–201.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sascha Schroeder, Jukka Hyönä, and Simon Liversedge. 2015. Developmental eye-tracking research in reading: Introduction to the special issue. *Journal of Cognitive Psychology*, 27:500–510.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Ekta Sood, Simon Tannert, Diego Frassinelli, Andreas Bulling, and Ngoc Thang Vu. 2020a. Interpreting attention models with human visual attention in machine reading comprehension. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 12–25, Online. Association for Computational Linguistics.

Ekta Sood, Simon Tannert, Philipp Mueller, and Andreas Bulling. 2020b. Improving natural language processing tasks with human gaze-guided neural attention. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Efstathios Stamatatos. 2011. Plagiarism detection using stopword n-grams. *Journal of the American Society for Information Science and Technology*, 62(12):2512–2527.

Shiyang Su and Mark L Davison. 2019. Improving the predictive validity of reading comprehension using response times of correct item responses. *Applied Measurement in Education*, 32(2):166–182.

Ece Takmaz, Sandro Pezzelle, Lisa Beinborn, and Raquel Fernández. 2020. Generating Image Descriptions via Sequential Cross-Modal Alignment Guided by Human Gaze. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4664–4677, Online. Association for Computational Linguistics.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

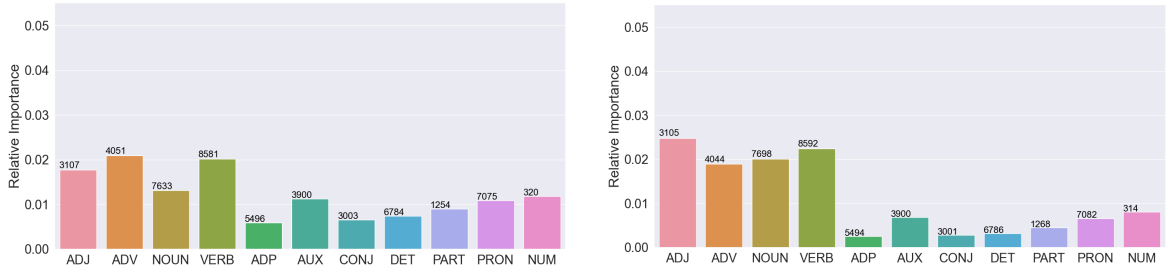
Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3).

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

A Additional Results



(a) Human Fixation (b) Model saliency

Figure 4: Relative importance of tokens with respect to word class in the GECO dataset. Relative importance is measured as relative fixation duration for humans (top) and as relative gradient-based saliency in the BERT model (bottom). This is the same figure as Figure 2 in the paper but it includes the number of instances per word class on top of the respective bar.

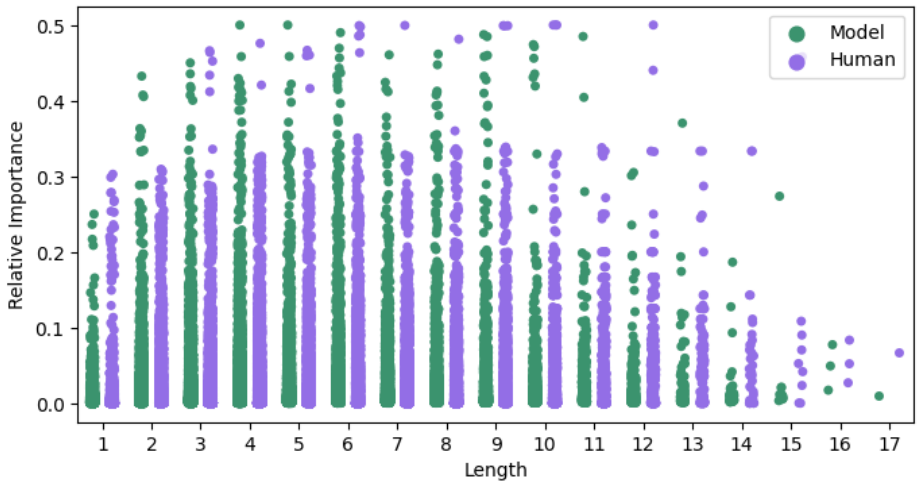


Figure 5: Relative importance values with respect to word length from human readers and from the BERT model for the GECO corpus.

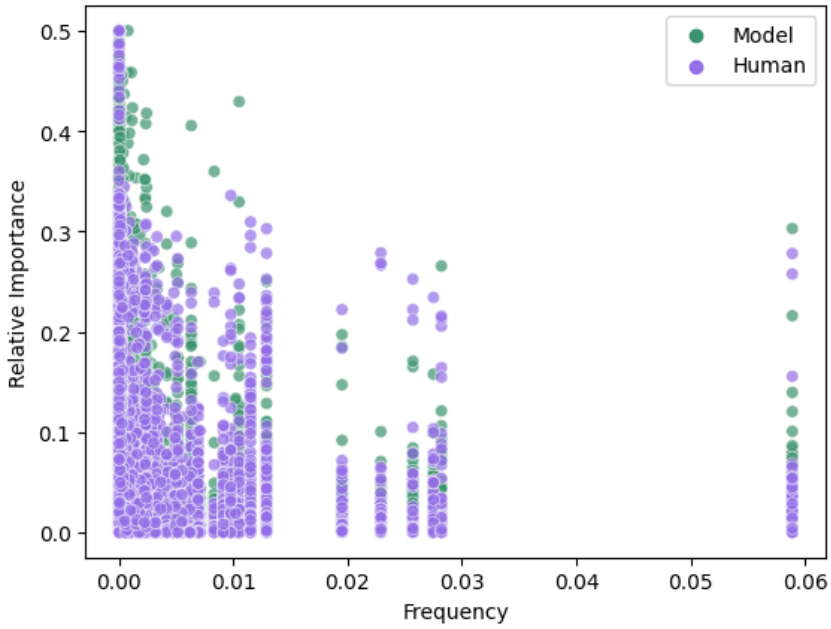


Figure 6: Relative importance values with respect to word frequency from human readers and from the BERT model for the GECO corpus.

Doing Good or Doing Right?
Exploring the Weakness of Commonsense Causal Reasoning Models

Mingyue Han[1], Yinglin Wang[2,*]
School of Information Management and Engineering,
Shanghai University of Finance and Economics, Shanghai, China
[1]mingyue.han@163.sufe.edu.cn, [2]wang.yinglin@shufe.edu.cn

Abstract

Pretrained language models (PLM) achieve surprising performance on the Choice of Plausible Alternatives (COPA) task. However, whether PLMs have truly acquired the ability of causal reasoning remains a question. In this paper, we investigate the problem of semantic similarity bias and reveal the vulnerability of current COPA models by certain attacks. Previous solutions that tackle the superficial cues of unbalanced token distribution still encounter the same problem of semantic bias, even more seriously due to the utilization of more training data. We mitigate this problem by simply adding a regularization loss and experimental results show that this solution not only improves the model's generalization ability, but also assists the models to perform more robustly on a challenging dataset, BCOPA-CE, which has unbiased token distribution and is more difficult for models to distinguish cause and effect.

1 Introduction

Supervised learning algorithms recklessly absorbing all the correlations found in training data is statistically correct but might have missed the point (Ahuja et al., 2020). Hence, recent work has focused more on spurious correlations in datasets in computer vision and NLP (Jia and Liang, 2017; McCoy et al., 2019). In inference tasks over natural language, spurious correlation has been identified a lot, such as lexical and grammatical constructs, word overlap, sentence length (Gururangan et al., 2018), and unbalanced token distribution (Poliak et al., 2018; Kavumba et al., 2019). COPA (Roemmele et al., 2011) is a natural language understanding task, which requires a system to choose either a cause or effect of a given story event. It is one of the natural language understanding tasks in SuperGlue benchmark (Wang et al., 2019). Pretrained language models gain a great improvement on COPA, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020). The recent state-of-the-art model on COPA, DeBERTa (He et al., 2020), reached a surprising accuracy of 98.4%. However, the complexity of causal reasoning and the requirements of world knowledge imply that the ability of causal reasoning in PLMs might be overestimated. It is worth exploring whether the models have acquired the ability of causal reasoning.

A Sample from development dataset
Premise: The woman banished the children from her property.
ask-for: "cause"
Alt1: The children hit a ball into her yard. × **(effect)**
Alt2: The children trampled through her garden. √ **(cause)**

Table 1: A challenging case where BERT predicts wrongly

We observe that 66.8% accuracy can be reached by a text semantic similarity model (Mulyar, 2020) based on BERT which is close to the performance (69.5%) of fine−tuning BERT on COPA training set. It indicates BERT is over-dependent on semantic similarity. Since the *cause* and *effect* of

*Corresponding Author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 151–157
August 1–6, 2021. ©2021 Association for Computational Linguistics

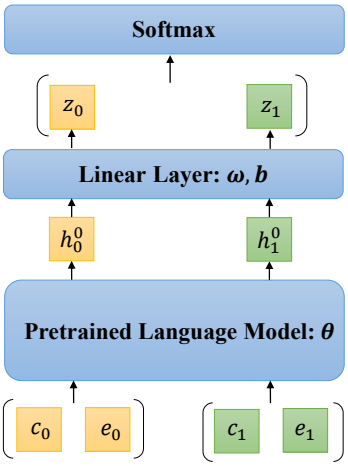


Figure 1: The general architecture of the PLMs on COPA task.

the same event often share the similar context, can PLMs really discriminate what we are asking for? A special case where BERT made mistakes on COPA development set in Table 1 seems to confirm our conjecture. BERT is more likely to fail in these challenging samples where the wrong alternative is the answer of its reverse question type. These investigations imply the models with satisfactory performance might have focused excessively on the topic semantic similarity instead of understanding cause and effect more finely. For this purpose, we design several probing experiments (Section 2) to verify our conjecture: (1) perturbation with distractors, (2) masking question type.

The main work on exploring bias in COPA is from Kavumba et al. (2019). They investigate unbalanced token distributions in correct answers in COPA training set and show that the good performance brought by BERT can be explained by its ability to exploit token distribution in alternatives. They augment the training set with a mirrored-COPA set to prevent the models from predicting with token distribution imprudently. However, we observed this improved model relies on semantic bias more seriously than the original PLMs. We further test the models on a new dataset, BCOPA-CE, which evaluates the ability of a system to distinguish the *cause* and *effect* and to reason without the clues of token distribution. For alleviating the semantic bias problem, we propose to add a regularization loss to the original objective (Section 3). Experimental results show that this solution is not only effective in our challenging test set, but improves the generalization ability of the model on the original test set. It also performs more robustly than the original PLMs in COPA-test *hard* set proposed by Kavumba et al. (2019).

In sum, our contributions are as follows:

(1) We explore the vulnerability of different COPA models by perturbing them with distractive alternatives. (2) We mitigate the weakness of COPA models by adding a regularization loss while maintaining their generalization ability. Our improved models also perform more robustly on the COPA-test hard set. (3) We introduce the BCOPA-CE dataset, which can evaluate the ability of a system to distinguish the cause and effect and to choose cause or effect under unbiased token distribution.

2 Probing Experiments

Unlike bias about token distribution or sentence length, indirect semantic cues cannot be analyzed statistically. We explore whether PLMs rely excessively on semantic similarity with special probing experiments. Firstly, we observe whether the model has dropped to a great extent if they see a distractive alternative, like a ***premise***. This distractor cannot be the correct answer, but it has a higher similarity score than the correct alternative. Moreover, inspired by Table 1, we investigate whether the model is aware of the question type during prediction. This is achieved by evaluating the model's performance while removing/masking the question type. We observe whether they still keep good performance without seeing the question type. We describe the model implementation details in Appendix A.

2.1 Exp1: Perturbation with Distractors

Model architecture: General PLMs assume that the first sentence and the second sentence describe a cause and an effect, respectively. For example, BERT take as input {cause, [SEP], effect}, which entails the question type in its formation. The general architecture in our experiment is shown in Figure 1. The shared parameters θ, ω, b are learned to classify each choice independently with the premise, where (c_i, e_i) is the i-th cause-effect pair, taking the first hidden vector in the final PLM layer:

$$h_i^0 = \theta(c_i, e_i) \quad (1)$$

yielding the logits for each cause-effect pair:

$$z_i = \omega^T h_i^0 + b \quad (2)$$

For training, we pass the logits $[z_0; z_1]$ through a softmax function to determine a probability

distribution and minimize the cross-entropy loss with the labels. For prediction, we choose the answer with the highest score by $i^* = argmax_{i\in\{0,1\}} z_i$. If we evaluate the trained models on ternary-choice test set, the prediction is then $i^* = argmax_{i\in\{0,1,2\}} z_i$.

Perturbation: We perturb models by adding a third choice, which does not affect human judgment. The "*premise*" is a good candidate since it is highly semantically related to itself while it cannot be the cause or effect of itself due to the non-reflexive trait of causality. We anticipate that the model will change its prediction when it meets the added choice. Meanwhile, we need to make sure that the performance drop is not from the increased difficulty of the problem since it becomes a ternary choice from a binary choice, hence we compare the results with a control experiment, where we add a choice randomly sampled from the COPA-test set.

- **COPA-random**: We control the difficulty of perturbation test by taking a wrong choice randomly sampled from the COPA-test set as the third alternative for each sample. We refer to COPA-random as "**Rand**" in Table 2.
- **COPA-premise:** we take the premise as the third alternative. We refer to COPA-premise as "**Prem**" in Table 2.

2.2 Exp2: Masking Question Type

As mentioned above, models are likely to ignore the question information (*cause* or *effect*, often share the same context) if they rely excessively on the semantic similarity. We mask the "*ask-for*" for each sample in COPA-test set by inputting the models with an arbitrary question type. The order of the alternative and the premise is determined by the question type. In masking setting, we randomly input [alternative; premise] or [premise; alternative] for each instance in spite of the question. In this way, half of the samples will keep the original question type, and the other samples get the wrong question type, which do not have the real correct answer. We observe whether these models still keep good performance without seeing the question type. If they do, the question type is ignored for the prediction of the models. We refer to this experimental setting as "**Mask**" in Table 2. The lower accuracy on "Mask" setting, the more robust the models are.

2.3 Baseline models

We conduct the aforementioned experiments with both traditional and SOTA COPA models.

- **CS:** Sasaki et al. (2017) handled the COPA task by statistically estimating causality scores using causal knowledge extracted from a corpus with causal templates.
- **PLMs:** We take BERT-large, RoBERTa-large, ALBERT-xxlarge-v1, and DeBERTa-large as baseline models (referred to as b-l, rb-l, alb, and db-l, respectively), and fine-tune them on the COPA-dev set, using the implementation from hugging face[1].
- **PLMs-aug (**b-l-aug, rb-l-aug, alb-aug and db-l-aug**):** PLMs are fine-tuned on BCOPA, a dataset with unbiased token distribution between the correct alternatives and the wrong alternatives proposed by Kavumba et al. (2019). The BCOPA dataset was constructed by mirroring the original training set with a modified premise.

2.4 Results and analysis

As is shown in Table 2, The CS method based on causal knowledge is the most robust system, barely affected by the added alternative. PLMs show different degrees of weakness when they are disturbed by the added alternative. The defensive

Model	Exp1: Perturbation			Exp2: Masking	
	Rand ↑	Prem ↑	Δ ↓	Test ↑	Mask ↓
CS	70.1	70.5	-0.4	70.8	61.1
b-l	59.3	11.6	47.6	69.5	69.0
b-l-aug	63.3	13.0	50.3	70.0	69.6
rb-l	83.3	66.7	16.6	86.3	82.8
rb-l-aug	85.6	65.7	19.9	87.3	83.5
alb	86.7	71.9	14.7	88.0	80.2
alb-aug	86.4	61.2	25.2	87.9	84.1
db-l	90.8	77.9	12.9	91.6	87.8
db-l-aug	91.1	78.9	12.2	91.8	88.8

Table 2. The accuracy of models in probing experiments. "↓" denotes a negative indicator (the lower, the better) and "↑" denotes a positive indicator (the higher, the better).

[1] The PLMs could be found at https://github.com/huggingface/transformers

A Sample in COPA-test set	New Samples in BCOPA-CE test set	
Premise: The accident was my fault. *ask-for:* "effect" *Alt1:* I felt guilty. √ *Alt2:* I pressed charges. ×	*Premise:* The accident was my fault. *ask-for:* "effect" *Alt1:* I felt guilty. √ *Alt2:*I was absent-minded. ×	*Premise:* The accident was my fault. *ask-for:* "cause" *Alt1:* I felt guilty. × *Alt2:* I was absent-minded. √

Table 3 The samples in COPA-test set and BCOPA-CE test set.

ability of BERT is the weakest, which is almost completely fooled by distractor and remains the original accuracy without seeing the questions. RoBERTa, ALBERT, and DeBERTa also drop 16.6%, 14.7%, 12.9% respectively compared with the performance of "**Rand**" setting. The fact that the systems perform worse on "**Prem**" (*premise* as a distractor) supports our hypothesis that PLMs have semantic similarity bias. This is because the *premise* is 100% similar to itself, being much more similar than a random distractor. For masking experiments, the theoretical accuracy of a perfectly robust model should be half of the chance-level (i.e., 50%) plus half of the original accuracy. The CS method achieves an accuracy of 61.1% and pays attention to the question type. On the contrary, PLMSs seem not to be aware of the question type and perform similarly without this information as original model setting. However, PLMs do not completely ignore the question type since they do not keep the same performance as the original test set.

We also investigate the robustness of the debiased methods of augmenting training data which focus on the unbalanced token distributions proposed by Kavumba et al. (2019). They suffer from the same issue even more seriously than the original PLMs except DeBERTa. This might be due to the fact that the models are more likely to capture the semantic similarity since each alternative pair in BCOPA appears twice.

3 Model-improving Method

3.1 BCOPA-CE Test

As is shown in Table 3, we introduce a balanced COPA test set, BCOPA-CE, by taking cause event and effect event as two alternatives for each premise. Specifically, for each premise of the 500 samples in COPA-test set, we generate one event manually which is a plausible answer to the opposite question type of the original sample, In the sample in Table 3, for the premise: "*The accident was my fault.*", we generate the *cause* of it: "*I was absent-minded.*", since the original question is asking for "*effect*". After this process, we obtain 500 triplets of <*premise*, *cause*, *effect*>. Then, we construct 1000 samples by giving two different questions (*cause* or *effect*) to each triplet. This guarantees the balanced token distribution between the correct and the wrong alternatives. The dataset generation details are described in Appendix B. Human evaluation has been conducted to ensure the quality of the new dataset in Appendix C.

3.2 Regularization Loss

We expect the model to make good choices while paying attention to the question type. For a sample in the COPA training set, the proposed loss includes two parts: The CrossEntropy loss and a regularization loss. The first part prompts the model to answer correctly given the question type. The extra regularization loss requires that a model should be neutral when it sees the opposite question type for the same premise and same alternatives, since neither alternative is the correct answer.

General PLMs take the first input sentence as the cause, and the second sentence as the effect. Mathematically, the logits of two input sentences in reverse cause-effect order should be as close as possible, even if one of two alternatives is semantically similar to the premise (the correct answer of the original question).

$$L = (1-\lambda) * L_{CE} + \lambda * L_{Reg} \quad (3)$$

$$L_{Reg} = \|z_0^r - z_1^r\|_2^2 \quad (4)$$

z_i^r is the logit of input $[e_i; c_i]$ computed by equation (1), which reverses the order of cause and effect of choice i. We set $\lambda = 0.01$ in all experiments corresponding to regularization loss.

3.3 Result and Analysis

Table 4 demonstrates the performance of our improved models on the COPA-test set, the BCOPA-CE set and the COPA-hard set. It's noted that the models with a regularization loss not only have improved performance on BCOPA-CE set,

Model	Test ↑	BCOPA-CE ↑	Δ ↓	Test-hard ↑
b-l	69.5	51.5	18.0	61.6
b-l-reg	**71.1**	**64.1**	**7.0**	63.6
b-l-aug	70.0	51.1	18.9	**69.7**
rb-l	86.3	73.0	13.3	83.1
rb-l-reg	**87.7**	**83.9**	**3.8**	84.5
rb-l-aug	87.3	69.2	18.2	**87.0**
alb	88.0	80.5	7.6	86.9
alb-reg	**89.4**	**86.7**	**2.7**	**88.6**
alb-aug	87.9	71.4	16.5	88.0
db-l	91.6	72.3	19.3	88.6
db-l-reg	**92.2**	**86.3**	**5.9**	89.7
db-l-aug	91.8	69.8	21.9	**90.5**

Table 4 The performance of PLMs and their variants on challenging set. Bold represents the best model setting in the same PLM.

but also perform better than the original PLMs on COPA-test set. Previous debiased models on token distribution perform worse than the original model, which is consistent with our conjecture that they amplify the semantic bias. Our solution also performs better on COPA-test-hard than the original PLMs, which has balanced token distribution as Kavumba et al. (2019) introduced. Regularization in our method considers debiasing token distribution as well, because we tend to stop the models from capturing any cues when it reverses the input order.

3.4 Error Analysis

We conduct an error analysis for the SOTA model, DeBERTa, using the run that is closest to the average of 20 runs. We give an example (the second row) from the BCOPA-CE dataset in Table 5 where DeBERTa predicts wrongly but the regularized DeBERTa model succeeds. Interestingly, both models make a correct prediction on the original sample (the first row) from COPA-test set, which indicates that the new alternative we generate perturbs the choice of the original DeBERTA model.

We calculate the word importance of all tokens in correct answer through erasure (Li et al., 2017). The importance score is computed by the relative difference in log likelihood on gold-standard labels while replacing the token with [MASK]. We observe two models predict correctly in this original sample but with different attention on tokens. As is shown in Figure 2, DeBERTa chooses Alt2 by focusing on "*He*" and "*spoke*", but DeBERTa-reg pays the most attention to

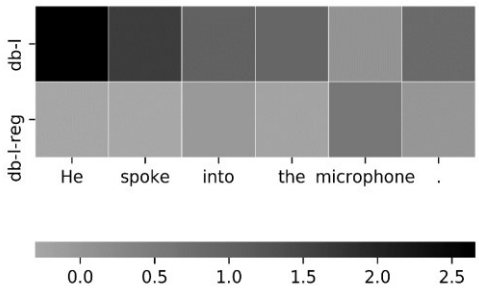


Figure 2: Heatmap of importance of each token in correct answer for the db-l model and db-l-reg model.

Original sample	*Premise:* The man's voice projected clearly throughout the auditorium. *Ask-for:* cause *Alt1:* He greeted the audience. × *Alt2:* He spoke into the microphone. √
New sample	*Premise:* The man's voice projected clearly throughout the auditorium. *Ask-for:* cause *Alt1:* Everyone heard him. × *Alt2:* He spoke into the microphone. √

Table 5 The case where DeBERTa is perturbed but regularized DeBERTa not.

"*microphone*", which is more in line with human causal intuition. When people make such inference, the causal relation between "*microphone*" and "*projected clearly throughout the auditorium*" should be more important than the co-reference relationship.

4 Conclusion

In this paper, we explore whether COPA models rely excessively on semantic similarity for prediction. We add the regularization loss to the training objective to alleviate this weakness. Results show that our solution is effective in our adversarial test, and improve the generalization ability and the robustness of models on previous COPA-hard dataset. Moreover, previous debiased models on token distribution rely on semantic bias more seriously than the original models, which reminds us if debiasing bring more other bias.

Acknowledgements

Special thanks to all annotators for their hard work. This work was supported by the National Natural Science Foundation of China (under Project No. 61375053). We thank Ming Wang, Jingshu Zhang, Dandan Li and the anonymous reviewers for their insightful comments and discussion.

References

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. 2020. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial Removal of Demographic Attributes from Text Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. *Deberta: Decoding-enhanced bert with disentangled attention.* arXiv:2006.03654.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2418–2428.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. 2019. When Choosing Plausible Alternatives, Clever Hans can be Clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.* arXiv: 1909.11942.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. *Understanding Neural Networks through Representation Erasure.* arXiv: 1612.08220.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *Roberta: A robustly optimized bert pretraining approach.* arXiv:1907.11692.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Andriy Mulyar. 2020. *AndriyMulyar/semantic-text-similarity: Zenodo Archiving Release*. Zenodo.

Timothy Niven and Hung-Yu Kao. 2019. Probing Neural Network Comprehension of Natural Language Arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis Only Baselines in Natural Language Inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, pages 90–95.

Shota Sasaki, Sho Takase, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui. 2017. Handling multiword expressions in causality estimation. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers.*

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *Superglue: A stickier benchmark for general-purpose language understanding systems.* arXiv:1905.00537.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, et al. 2020. *HuggingFace's Transformers: State-of-the-art Natural Language Processing.* arXiv: 1910.03771.

Appendix

A Implementation Details

PLMs: We randomly split the training set (COPA-dev, or BCOPA) into training set and development set with a ratio of 9:1, and finetune our model up to 20 epochs by implementing an early-stopping strategy with a patience of 5 epochs and using AdamW optimizer. We run 20 different random seeds for each supervised model and report the mean of the non-degenerate runs for each model, which have higher than 80% of accuracy in the training set as in previous work (Niven and Kao, 2019).

CS: We reproduce the preprocessing of their work and achieve 70.8% accuracy, which is slightly lower than the reported accuracy of 71.4%.

All parameters are learned from the development set by manual tuning. The best-performing parameter is determined by the accuracy of the model in the development set. The final parameters in our experiments are shown in Table 6.

Model	LR	BS	WD	WP	λ
b-l	1e-4	32	0.01	0.1	-
b-l-aug					
b-l-reg	8e-5				0.01
rb-l	8e-6	32	0.01	0.06	-
rb-l-aug					
rb-l-reg	1.2e-5				0.01
alb	1.1e-4	48	0	0	-
alb-aug					
alb-reg	6e-5				0.01
db-l	5e-6	32	0.01	0.06	-
db-l-aug					
db-l-reg	1e-5				0.01

Table 6. The best Batch Size (BS), Learning Rate (LR), Warm up rate (WP)， and Weight Decay value (WD) we used in our experiments.

B Construction Details of BCOPA-CE

We asked five fluent English speakers who have background knowledge of NLP to create the new alternative with the specific guidelines. We instructed creators with requirements of sentence length, overlap rules, and expressions similar to Kavumba et al. (2019).

C Human Evaluation on BCOPA-CE

We have 1000 samples in BCOPA-CE set, which consist of. 500 samples whose answers are same with original COPA-test set (the left sample in the second column in Table 3, referred to as COPA-CE-ori) and 500 samples whose answers are the choices that we generate (the right sample in the second column in Table 3, referred to as COPA-CE-opp). To ensure the quality of generated dataset, we conduct a quality evaluation with two questions:

- ***Q1****: Are the instances in* BCOPA-CE dataset *comparable in difficulty to the COPA-test instances?*
- ***Q2****: Is the new alternative we collect plausible for the opposite question type?*

	COPA-test	COPA-CE-ori	COPA-CE-opp
Accuracy	0.980	0.990	1.000
Fleiss' Kappa	0.919	0.893	0.890

Table 7: Human evaluation result of generated dataset.

We evaluate the accuracy of human on both COPA-CE-ori dataset and COPA-CE-opp dataset to answer the Q1 and evaluate the human performance on COPA-CE-opp set for Q2. The COPA-CE-opp set changes the question type and takes the generated event as gold answers, hence it can be evaluated for the plausibility of generated alternatives. We asked 9 people to make choices, each group of 3 people for one dataset. We determine the final choice by majority voting. The inter-annotator agreement is calculated by Fleiss' Kappa. As is shown in Table 7, the BCOPA-CE set has comparable difficulty with COPA-test. The performance on COPA-CE-opp shows that the new alternatives we create are plausible.

AND does not mean OR: Using Formal Languages to Study Language Models' Representations

Aaron Traylor
Dept. of Computer Science
Brown University

Roman Feiman
Dept. of Cognitive, Linguistic, and Psychological Sciences
Brown University

Ellie Pavlick
Dept. of Computer Science
Brown University

{aaron_traylor, roman_feiman, ellie_pavlick}@brown.edu

Abstract

A current open question in natural language processing is to what extent language models, which are trained with access only to the *form* of language, are able to capture the *meaning* of language. In many cases, meaning constrains form in consistent ways. This raises the possibility that some kinds of information about form might reflect meaning more transparently than others. The goal of this study is to investigate under what conditions we can expect meaning and form to covary sufficiently, such that a language model with access only to form might nonetheless succeed in emulating meaning. Focusing on propositional logic, we generate training corpora using a variety of motivated constraints, and measure a distributional language model's ability to differentiate logical symbols ($\neg$, $\wedge$, $\vee$). Our findings are largely negative: none of our simulated training corpora result in models which definitively differentiate meaningfully different symbols (e.g., $\wedge$ vs. $\vee$), suggesting a limitation to the types of semantic signals that current models are able to exploit.

1 Introduction

A current open question in natural language processing is to what extent language models (LMs; neural networks trained to predict the likelihood of word forms given textual context) are capable of truly understanding language. Bender and Koller (2020) argue that, since such models are trained exclusively on the *form* of language, they cannot possibly learn the *meaning* of language. We argue that the question of whether language models can learn meaning cannot be settled *a priori*. While language models only have direct access to form, linguistic form often correlates with meaning. The strength of the correlation varies across both different aspects of language and different tests of linguistic competence. While several intuitive tests of understanding (e.g., demonstrating knowledge of the word *dog* by identifying pictures of dogs) are out of scope for LMs, many tasks which NLP aspires to solve (e.g., question answering, machine translation) operate entirely on natural language input and output. Thus, a relevant question is whether models which operate only on the forms of language can nonetheless learn to differentiate meanings.

Our goal is to focus on a tractable subproblem in order to improve our intuitions about the types of distributional signals that LMs can use to extract information relevant to meaning. We simulate a language modeling setup using propositional logic, in which we can naturally operationalize *form* to be strings of symbols in the language and *meaning* to be truth conditions. We define the *semantic transparency* of a text-only training corpus to be the degree to which an LM trained on that corpus learns to differentiate between aspects of form that affect truth conditions and aspects of form that do not. We have two primary research questions. First, what constraints on corpus generation produce greater semantic transparency? And second, are any such constraints sufficient for an LM to adequately differentiate meanings?

2 Experimental Design

2.1 Dataset Generation

We consider the *form* of a sentence to be simply the observed, syntactically-valid strings of characters and the *meaning* to be the truth conditions. Propositional logic is a simple language in which we can characterize both form and meaning. We use the grammar in Table 1, with standard semantics.

We focus our analysis on whether the representations of logical operators ($\wedge$, $\vee$, $\neg$) are influenced by distributional patterns that go beyond their superficial syntactic similarity evident in the grammar. That is, if a trained LM identifies that the meanings

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 158–167
August 1–6, 2021. ©2021 Association for Computational Linguistics

$S \rightarrow$	$(S \wedge S) \mid (S \vee S) \mid (\neg S) \mid (\texttt{sym})$
$\wedge \rightarrow$	$\wedge_1 \mid \wedge_2 \cdots \mid \wedge_K$
$\vee \rightarrow$	$\vee_1 \mid \vee_2 \cdots \mid \vee_L$
$\neg \rightarrow$	$\neg_1 \mid \neg_2 \cdots \mid \neg_M$
$\texttt{sym} \rightarrow$	$\texttt{sym}_1 \mid \texttt{sym}_2 \cdots \mid \texttt{sym}_N$

Table 1: Propositional logic grammar.

of $\wedge_1 \cdots \wedge_k$ are identical to one another, and different from the meanings of $\vee_1 \cdots \vee_l$, we expect the embeddings for the $\wedge_i$ to be more similar to one another than they are to any of the $\vee_i$ or the $\neg_i$. We consider a corpus to be *semantically transparent* if an LM trained on the corpus learns semantically-clustered representations of the logical operators.

We generate four different training corpora, motivated by different assumptions one might make about how natural language corpora arise. These constraints are as follows, ordered roughly from weakest to strongest:

1. Syntactic Constraint. Speakers only generate sentences which are syntactically well-formed (that can be parsed by a syntactic parser). Here, this amounts to sampling from the grammar without additional constraints.

2. Truthfulness Constraint. Speakers of the language are constrained to generate sentences that are true in some context, i.e., that evaluate to `True` in at least one possible world. To implement this, we again sample from the grammar but additionally check with a satisfiability checker and omit sentences which are not satisfiable. E.g., $(\texttt{sym}_1 \wedge (\neg(\texttt{sym}_1)))$ would not appear.

3. Informativity Constraint. Speakers generate sentences not just to state true facts, but to provide listeners with information about a particular state of affairs. To simulate such a constraint, we randomly sample a set of "target worlds" T and a set of "alternative worlds" A such that $T \cap A = \emptyset$. We then generate the shortest sentence s such that s is true in every world in T and s is false in every world in A. We experiment with several sizes of T and A, but report only on $|T| = |A| = 2$ as this provides the right balance of contextual diversity. See Appendix for additional discussion.

4. Explicit Grounding. We consider a setting in which speakers explicitly dictate the full state of affairs, without ambiguity. This is not intended as a realistic model of how corpora are generated, but rather to provide an upper bound on semantic transparency by giving models a corpus in which form is perfectly correlated with meaning. We generate this corpus in the same way as the Truthfulness corpus, but append an explicit marker of the truth values[1] of the variables in the sentence, e.g.: $(\texttt{sym}_1 \wedge (\neg(\texttt{sym}_2)))$ `<sep>` $\texttt{sym}_1$ `T` $\texttt{sym}_2$ `F`.

Sampling Parameters. Each dataset consists of 100K training and 1K validation sentences. We set the number of non-reserved symbols (N in the above grammar) to 5,000, and the number of "synonyms" of each logical symbol (K,L,M) to be 5. Thus, a sentence in one of our datasets might look like $(\texttt{sym}_1 \wedge_3 (\neg_4(\texttt{sym}_{85})))$, and would be true if and only if $\texttt{sym}_1$ is true and $\texttt{sym}_{85}$ is false [2].

We generate sentences using a probabilistic context-free grammar with the rules shown above. The tree depth d of a generated sentence is controlled by a parameter γ such that $P(d|d-1) = \gamma^d$. The number of unique variables in a sentence[3] is sampled from a non-zero Poisson distribution parameterized by λ. We set $\lambda = 2$ and $\gamma = .85$ in the reported experiments, but don't find parameter choice affects our conclusions. Note that the Informativity dataset is generated deterministically, and thus sampling parameters do not apply and sentences in that dataset are shorter. Dataset statistics and data generation parameter sensitivity are in the Appendix.

2.2 Models and Training

We consider LSTM and Transformer LMs of differing sizes, shown in Table 2. Each model is trained on one of the above four datasets until convergence on the associated validation set using early stopping with a patience of 15 epochs. The LMs were implemented in PyTorch (Paszke et al., 2019) and took roughly 5 hours to converge on TitanV, TitanRTX, and QuadroRTX GPUs [4]. We randomly initialize the embedding layer. Hyperparameter details can be found in the Appendix. We train 5 random restarts of each setting. Due to the regular nature of our synthetic data, we found larger mod-

[1] Sampled from the set of satisfying variable assignments.

[2] We began by experimenting with many different dataset sizes and vocab counts. However, we did not find that models behaved differently on larger datasets and so focused on the smaller ones for convenience. See Appendix for results with different model sizes.

[3] We set a maximum number of variables per sentence in order to bound the number of possible variable assignments.

[4] Code publicly available at https://github.com/attraylor/semantic-transparency-code.

Model	Syntactic	Truthfulness	Informativity	Grounded
Small LSTM (192K)	21.2 / 87.7 / 87.7	17.6 / 88.7 / 88.6	21.5 / 99.6 / 99.5	21.2 / 87.5 / 87.5
Medium LSTM (545K)	17.6 / 90.2 / 90.1	17.5 / 89.6 / 89.5	20.9 / 99.9 / 99.8	8.3 / 89.3 / 86.8
Small Trans. (311K)	11.8 / 86.9 / 84.6	12.4 / 87.2 / 85.4	21.7 / 98.4 / 98.2	10.3 / 86.2 / 83.1
Medium Trans. (377K)	11.4 / 91.3 / 90.6	9.9 / 92.0 / 91.3	18.1 / 99.5 / 99.5	9.1 / 91.7 / 89.8

Table 2: Summary of language modeling performance. For each model, on each training dataset, we report **PPL / %Syn / %Sem** where PPL is the perplexity on heldout data (drawn from the same distribution as the training corpus), %Syn is the percentage of generated sentences that are syntactically well formed (i.e., parseable), estimated on a set of 1,000 generations sampled from the trained model, and % Sem is the percentage of generated sentences that are semantically well formed (i.e., satisfiable), estimated on the same set of 1,000.

els overfit the training data quickly, and thus focus on smaller models.

3 Results and Discussion

Language Modeling Performance. We first sanity check that the trained models indeed function as LMs before evaluating the lexical representations. We compute the models' perplexity on heldout data. However, since perplexity is not comparable across conditions (since each constraint leads to differently distributed corpora) we also sample 1,000 generated sentences from each model and compare by measuring whether the sentences are 1) syntactically well-formed (i.e., parseable) and 2) semantically well-formed (i.e., satisfiable). Even in the case of models trained with the Syntactic constraint, as seen in Table 2, most of the sentences produced are nonetheless satisfiable. We see no difference between the Syntactic, Truthfulness, and Explicit Grounding conditions on these metrics. (The Informativity numbers are likely higher due to the shorter sentences that result from that generative process.) The fact that models trained only on satisfiable sentences nonetheless generate sentences which do not abide by such constraints suggests the models fail to encode less overt distributional patterns, which depend, for example, on recognizing abstract relations such as "sameness" of symbols in order to recognize violations (e.g., (A $\wedge$($\neg$ A)). The failure to capture such properties of the data even in this simplified setting might have negative implications for the models' ability to infer abstract semantic relationships from more complex natural language corpora.

Representations of Logical Symbols. Again, our first question is: What constraints on corpus generation yield the greatest amounts of semantic transparency? We quantify this by measuring how well the embeddings learned by the trained LMs correspond to our truth-theoretic notions of semantic equivalence: e.g., are $\wedge_1$ and $\wedge_2$ more similar to one another than $\wedge_1$ and $\vee_1$? We use a nearest neighbors probing classifier to evaluate whether models distinguish the operators at the lexical level. We run k-fold cross validation, in each iteration choosing one symbol per class (i.e., one $\wedge$, one $\vee$, one $\neg$) as the class exemplars, and then classifying the remaining points using cosine similarity. We set k to 125, so that we observe every symbol combination as exemplars. We report accuracy averaged across folds and random restarts.

Probing classifier results are shown in Figure 1. Figure 2 shows an embedding visualization for one model (Medium Transformer). We find that training on the Syntactic and on the Explicit Grounding dataset leads to the least and the most distinguishable operators respectively for all models, and the other conditions end up between these values.

These results address our first question: there is some difference in semantic transparency between differently constrained datasets. Interestingly, the Transformer models perform better in the Truthfulness condition than in the Syntactic condition, which the LSTMs fail to differentiate. This suggests that, even if it does not necessarily manifest in the models' generations (Table 2), the Transformer architecture may nonetheless be capable of picking up on some of the more abstract distributional patterns via which syntax and semantics are correlated. Further work on larger models would be required to explore this in depth.

In addition, we observe little difference between the quality of the representations learned in the Informativity condition and those learned in the Truthfulness condition; one exception might be in the Medium LSTM, though we cannot confirm that this difference is robustly reproducible. Thus,

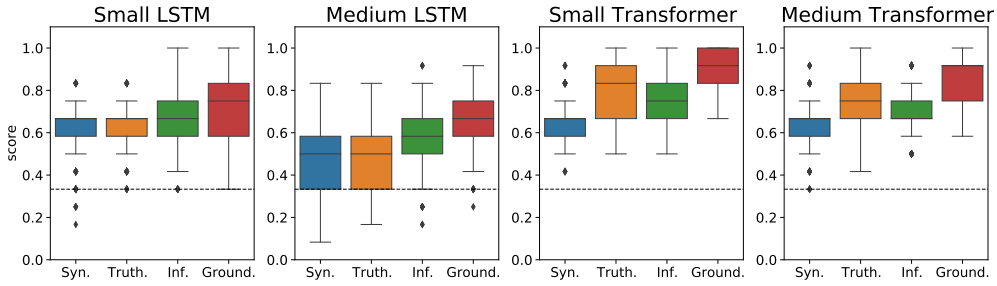


Figure 1: Each value in this graph represents average classification score across 125 iterations of a simple nearest neighbor probing classifier averaged across 5 random seeds of the model (625 accuracy numbers per box and whiskers plot). The dotted line is random chance / maximum class accuracy (33%).

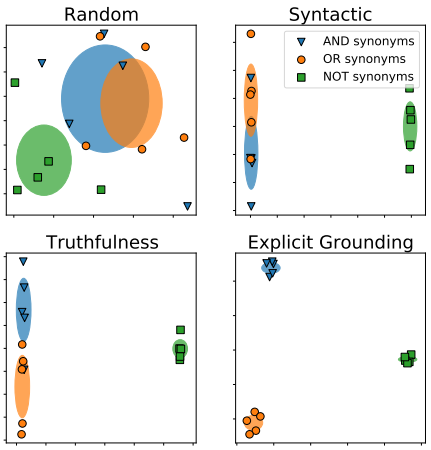


Figure 2: PCA of the representations created by the Medium Transformer model.

based on our experiments, there is no evidence that Informativity alone yields greater semantic transparency. However, we note that the experimental setup for Informativity is not directly comparable to the others (e.g., sentences are shorter and less diverse than in Truthfulness) and thus further study would be needed to make strong claims, positive or negative.

Finally, we note that in nearly all cases, models are able to differentiate $\neg$ from the other operators, likely because it is a unary operator and thus syntactically different from the binary operators. Thus the difference in accuracy is almost entirely due to whether the representations of $\wedge$ and $\vee$ are differentiated (as shown in Figure 2). This gives a negative answer to our second question concerning whether any constraints are sufficient for an LM to adequately differentiate meaning. Apart from the Small Transformer on the Explicit Grounding condition, none of the models can completely distinguish between symbols that are similar in form but different in meaning.

4 Related Work

It is an open question whether neural models can learn abstract functions (Marcus, 2001). Our work builds upon a large body of research intended to probe which aspects of language and meaning are being captured by large LMs. Most closely related is work that assesses whether models can perform symbolic reasoning about language (Kassner et al., 2020) e.g., quantifiers or negation (Talmor et al., 2020; Ettinger, 2020; Kassner and Schütze, 2020; Wang et al., 2018) or by measuring the systematicity of models' inferences (Goodwin et al., 2020; Kim and Linzen, 2020; Yanaka et al., 2020; Warstadt et al., 2019). Such work has tended to find that LMs reason primarily contextually as opposed to abstractly. Our evaluation method– which asks whether word embeddings cluster according to their truth-conditional meaning– is related to recent work which defines text-only models as "grounded" if the learned embedding space is isomorphic to the similarity function defined over a ground-truth meaning representation (Merrill et al., 2021). More distantly related is work on LMs' ability to reason about numbers (Wallace et al., 2019) or perform multi-hop reasoning (Yang et al., 2018). Prior work that examines neural networks' ability to perform logical reasoning is superficially related (Evans et al., 2018). In this way, our work builds on past work that uses synthetic rather than natural language datasets in order to probe model behavior in the absence of confounds. Notable examples are SCAN for measuring compositionality and generalization (Lake and Baroni, 2018) and Kassner et al. (2020) which investigates LM knowledge acquisition and fact memorization using a synthetic dataset of entity-relation tuples.

5 Conclusion

Using propositional logic corpora to simulate a controlled language modeling setting, we ask: 1) Do properties of the training corpus affect LMs' abilities to differentiate the meanings of logical operators? and 2) Do any training corpora lead to models that differentiate these meanings to a satisfactory degree? Our results imply a positive answer to (1): Models trained on corpora generated with different constraints appear to perform differently at the task of separating $\wedge$ from $\vee$. However, these differences are a function of both data and model. For example, the Transformer architecture seems better able to learn from weaker signal (corpora generated only with a Truthfulness constraint), while LSTMs require more explicit signal (direct access to truth values). On question (2), our results are largely negative for the syntactically similar operators. Even the most semantically transparent training data did not enable models to separate the representations of symbols with similar form but different meaning. Only the Small Transformer trained on the Explicit Grounding condition can perfectly differentiate $\wedge$ from $\vee$ at the lexical level, despite the task's controlled nature. However, every model did separate $\neg$ from both $\wedge$ and $\vee$, illustrating how syntactic differences can support differentiation of meaning.

Overall, we contribute a novel framework, based on syntax and semantics of propositional logic, via which we can explore questions of the linguistic capabilities and weaknesses of neural LMs. Our experiments represent a first step in this line of work, but further work is needed to fully appreciate the implications of these results in natural language settings, in particular, how closely the constraints explored here mirror real corpora, and how such learning is influenced by noise and ambiguity found in human language. One specific limitation of our experiments is that we constrain our analysis to the lexical representations– i.e., we assume that differences between the meanings of $\wedge$ and $\vee$ should be encoded in the lexicon, via context-invariant type embeddings. While this assumption is commonplace in formal semantics, neural LMs open the possibility of alternative representations of lexical and compositional semantics. Our results do not rule out the possibility that the relevant semantic distinctions are encoded elsewhere in the model, above the lexical layer. However, we take the combination of the lexical probing results and LM generation results as suggestive but not confirmational evidence of a more general negative finding.

6 Acknowledgements

We would like to thank Najoung Kim and the participants of the NALOMA 2020 Workshop for their thoughtful feedback on early versions of this work. This work was supported by IARPA under the BETTER program, contract number 19051600004.

References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. 2018. Can neural networks understand logical entailment? In *International Conference on Learning Representations*.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969. Association for Computational Linguistics.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818. Association for Computational Linguistics.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Gary F Marcus. 2001. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Will Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form:what will future language models understand? *TACL*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

7 Appendix

7.1 Dataset generation parameters

There are several parameters involved in the creation of our synthetic propositional logic datasets:

- Number of sentences in the training set
- Number of unique non-reserved variables (N)
- Number of each operator (K, L, M)
- Sentence depth parameter (γ)
- Poisson distribution parameter for unique non-reserved variables in sentence (λ)

In comparison to dataset sizes for large language models in modern natural language processing, the dataset size (100k training examples) and vocabulary size (5k symbols + 5 of each operator) of our main experimental results (Figure 1) are rather small. We sought to determine whether our choice for dataset size and non-restricted variable count greatly changed the final results– do our conclusions change based on these parameters? We trained models on different variations of our initial parameters.

First, we swept across training set sizes (20k, 100k, and 500k examples) and number of symbols (500, 5k, 50k) while holding all other parameters constant (γ = .85, λ = 2, K, L, M = 5). We used the Medium Transformer model, which performed the best across our four models, and observed the results of the probing classifier on the embeddings after training separately on each model.

The results of the above sweep are shown in Figure 3. We do not find that the models perform dramatically differently on any of the datasets when dataset size and number of non-reserved symbols are varied.

We also experimented with changing the number of operator synonyms (e.g. $\wedge_1, \wedge_2, ...\wedge_K$) We experimented with three different sizes– (K, L, M) = 5, 25, 100– for each of our 4 datasets. Those results are shown in Figure 5, and average frequency is shown in Table 3. We found that adding additional synonyms of each operator hurt performance– likely because adding additional synonyms of $\wedge$ and $\vee$ made generalization more challenging, causing the models' performance to drop.

In a set of earlier experiments, to choose the sentence depth (γ) and Poisson distribution (λ) parameters, we hyperparameter searched on the Explicit Grounding condition across three values of each (nine datasets in total). Specifically, we tested $\lambda = 2, 3, 5$ and $\gamma = .7, .8, .85$. We then trained the transformer model once on each of the nine datasets, and the results are shown in Figure 6. We chose $\lambda = 2$ and $\gamma = .85$.

K, L, M	Syn.	Tru.	Inf.	Grd.
5	49.7k	49.2k	16.3k	49k
25	9.94k	9.84k	3.25k	9.81k
100	2.49k	2.46k	0.81k	2.45k

Table 3: Average count of each operator across each of the datasets.

7.2 Informativity dataset information

We tested different settings of $|T|$ (number of target worlds) and $|A|$ (number of alternative worlds). For $|T| = 1, |A| = 1$, the best choice of s will always be a single $\texttt{sym}$ or its negation. For example, with variables $\texttt{sym}_1, \texttt{sym}_2$, we might sample max variables = 2 and thus $T = (\texttt{sym}_1 = \texttt{T}, \texttt{sym}_2 = \texttt{F}), A = (\texttt{sym}_1 = \texttt{F}, \texttt{sym}_2 = \texttt{F})$. The shortest sentence would then be $\texttt{sym}_1$, as it sufficiently distinguishes T from A. However, with $|T| = 1, |A| = 2$, we might generate $T = (\texttt{sym}_1 = \texttt{T}, \texttt{sym}_2 = \texttt{F}), A = ((\texttt{sym}_1 = \texttt{F}, \texttt{sym}_2 = \texttt{F}), (\texttt{sym}_1 = \texttt{T}, \texttt{sym}_2 = \texttt{T}))$. Now the shortest sentence that can be generated is $(\texttt{sym}_1 \wedge_1 \neg_1(\texttt{sym}_2))$.

$|T| = 1, |A| = 2$ and $|T| = 2, |A| = 1$ result in sentences that are both short and structurally nearly identical, although inverted. This is due to the truth conditions allowed by each operator. We generate the datasets for each combination and report the results in Table 4. We excluded these datasets because of the simplicity and similarity of the sentences. We found that $|T| = 2, |A| = 2$ allows for sentences that are much more varied.

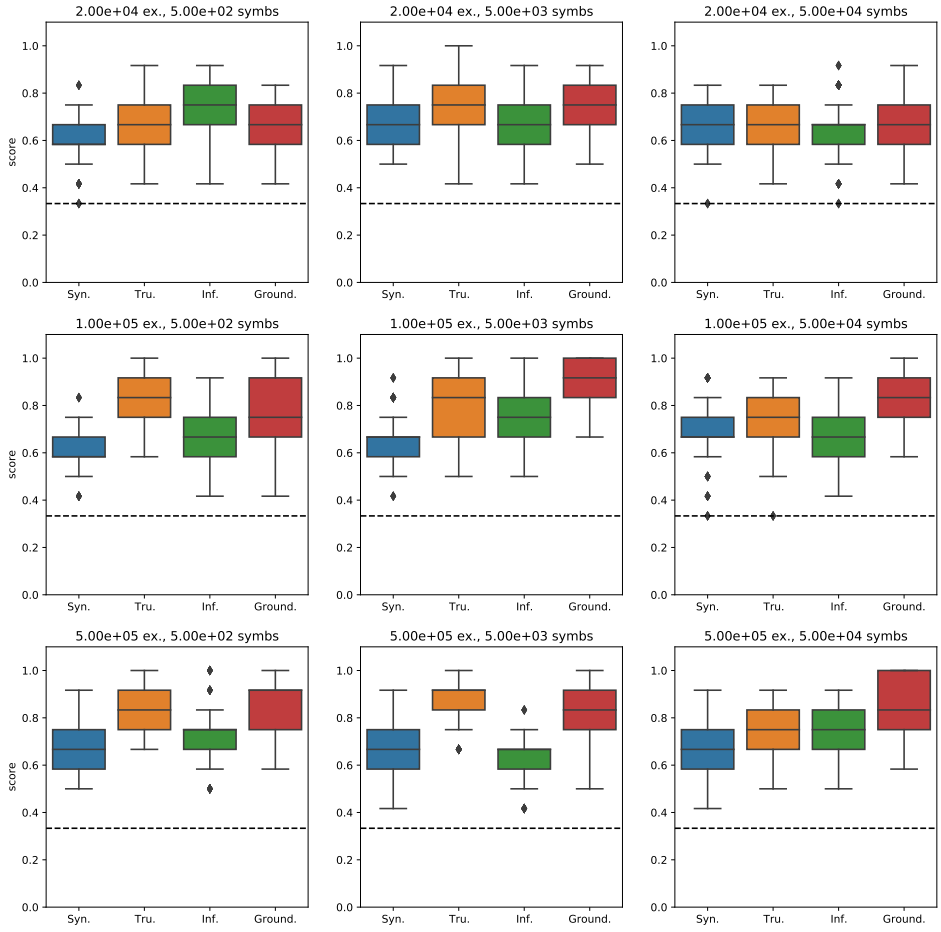


Figure 3: Average probing classifier score across example count / number of unique non-variable symbols for the Medium Transformer model.

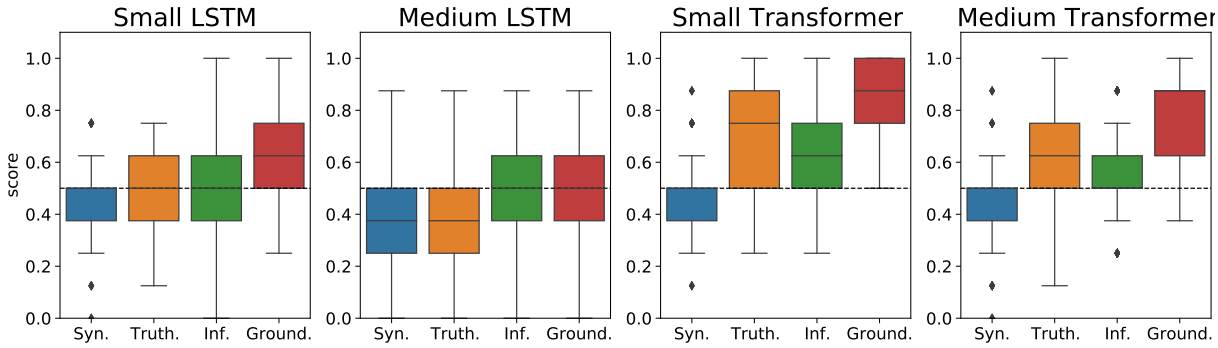


Figure 4: This graph contains the same experiments as Figure 1, but is only the accuracy on $\wedge$ and $\vee$, excluding the results of the negation operator.

Inform. 1T/1A		Inform. 1T/2A		Inform. 2T/1A	
Sent.	Count	Sent.	Count	Sent.	Count
a	4523	$(a \wedge b)$	27047	$(a \vee b)$	27236
$\neg(a)$	4460	$(a \wedge \neg(b))$	21474	$\neg((a \wedge b))$	21392
		$\neg((a \vee b))$	21338	$(a \vee \neg(b))$	21260
		$(\neg(a) \wedge b)$	21061	$(\neg(a) \vee b)$	21045
		$\neg(a)$	4544	a	4559
		a	4536	$\neg(a)$	4508

Table 4: All sentences generated for the first three Informativity datasets fell into one of these templates. Arbitrary symbols are replaced with a and b. This distinction happens because of the truth conditions that are allowed by the $\wedge$ and $\vee$ operators.

Dataset	Sent. Len.	Average `sym` count	Average op count	Average Unique `syms`
Syntactic	28.51	6.19	7.44	2.27
Truthfulness	28.25	6.14	7.37	2.33
Inform. (2T/2A)	10.92	2.83	2.70	2.20
Expl. Ground	34.06	8.51	7.40	2.33

Table 5: Averaged statistics per sentence for the different datasets (training sets). All datasets are 100K training examples and 1k heldout examples.

Model	LR	symb dim	hidden dim	# heads	# layers	dropout
Small LSTM	.0001	4	32		1	0.0
Medium LSTM	.0001	32	64		2	0.2
Small Transformer	.0001	4	32	2	4	0.0
Medium Transformer	5e-05	32	128	4	4	0.2

Table 6: Hyperparameters for each model.

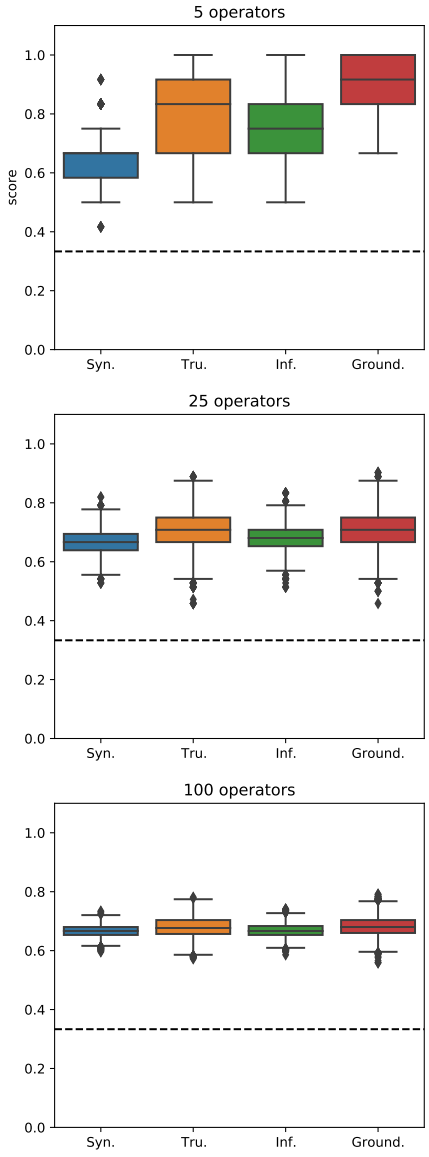


Figure 5: Sweep across number of operators using the Medium Transformer model.

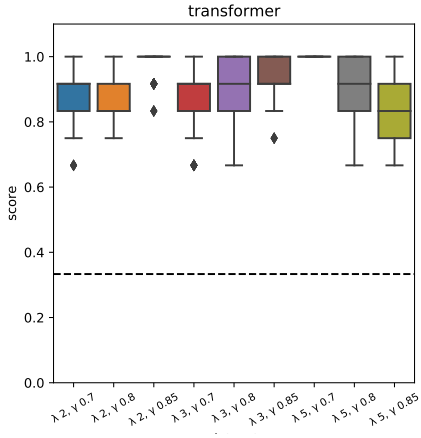


Figure 6: Sweep across λ and γ values for the Explicit Grounding dataset using a Transformer model.

Enforcing Consistency in Weakly Supervised Semantic Parsing

Nitish Gupta[*]
University of Pennsylvania
nitishg@seas.upenn.edu

Sameer Singh
University of California, Irvine
sameer@uci.edu

Matt Gardner
Allen Institute for AI
mattg@allenai.org

Abstract

The predominant challenge in weakly supervised semantic parsing is that of *spurious programs* that evaluate to correct answers for the wrong reasons. Prior work uses elaborate search strategies to mitigate the prevalence of spurious programs; however, they typically consider only one input at a time. In this work we explore the use of consistency between the output programs for related inputs to reduce the impact of spurious programs. We bias the program search (and thus the model's training signal) towards programs that map the same phrase in related inputs to the same sub-parts in their respective programs. Additionally, we study the importance of designing logical formalisms that facilitate this kind of consistency-based training. We find that a more consistent formalism leads to improved model performance even without consistency-based training. When combined together, these two insights lead to a 10% absolute improvement over the best prior result on the Natural Language Visual Reasoning dataset.

1 Introduction

Semantic parsers map a natural language utterance into an executable meaning representation, called a logical form or program (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005). These programs can be executed against a context (e.g., database, image, etc.) to produce a denotation (e.g., answer) for the input utterance. Methods for training semantic parsers from only (utterance, denotation) supervision have been developed (Clarke et al., 2010; Liang et al., 2011; Berant et al., 2013); however, training from such weak supervision is challenging. The parser needs to search for the correct program from an exponentially large space, and the presence of *spurious programs*—incorrect representations that evaluate to the correct denotation—greatly hampers learning. Several strategies have been proposed to mitigate this issue (Guu et al., 2017; Liang et al., 2018; Dasigi et al., 2019). Typically these approaches consider a single input utterance at a time and explore ways to score programs.

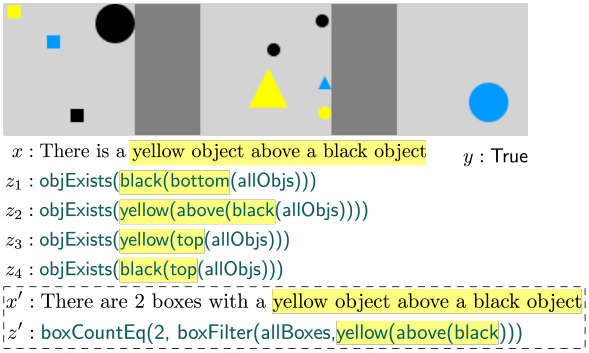


Figure 1: Utterance x and its program candidates z_1-z_4, all of which evaluate to the correct denotation (True). z_2 is the correct interpretation; other programs are *spurious*. Related utterance x' shares the phrase *yellow object above a black object* with x. Our consistency reward would score z_2 the highest since it maps the shared phrase most similarly compared to z'.

In this work we encourage consistency between the output programs of related natural language utterances to mitigate the issue of spurious programs. Consider related utterances, *There are two boxes with three yellow squares* and *There are three yellow squares*, both containing the phrase *three yellow squares*. Ideally, the correct programs for the utterances should contain similar sub-parts that corresponds to the shared phrase. To incorporate this intuition during search, we propose a consistency-based reward to encourage programs for related utterances that share sub-parts corresponding to the shared phrases (§3). By doing so, the model is provided with an additional training signal to distinguish between programs based on their consistency with programs predicted for related utterances.

[*] Work done while interning with Allen Institute for AI.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 168–174
August 1–6, 2021. ©2021 Association for Computational Linguistics

We also show the importance of designing the logical language in a manner such that the ground-truth programs for related utterances are consistent with each other. Such consistency in the logical language would facilitate the consistency-based training proposed above, and encourage the semantic parser to learn generalizable correspondence between natural language and program tokens. In the previously proposed language for the Natural Language Visual Reasoning dataset (NLVR; Suhr et al., 2017), we notice that the use of macros leads to inconsistent interpretations of a phrase depending on its context. We propose changes to this language such that a phrase in different contexts can be interpreted by the same program parts (§4).

We evaluate our proposed approaches on NLVR using the semantic parser of Dasigi et al. (2019) as our base parser. On just replacing the old logical language for our proposed language we see an 8% absolute improvement in consistency, the evaluation metric used for NLVR (§5). Combining with our consistency-based training leads to further improvements; overall 10% over the best prior model, reporting a new state-of-the-art on the NLVR dataset.

2 Background

In this section we provide a background on the NLVR dataset (Suhr et al., 2017) and the semantic parser of Dasigi et al. (2019).

Natural Language Visual Reasoning (NLVR) dataset contains human-written natural language utterances, where each utterance is paired with 4 synthetically-generated images. Each (utterance, image) pair is annotated with a binary truth-value denotation denoting whether the utterance is true for the image or not. Each image is divided into three *boxes*, where each box contains 1-8 *objects*. Each object has four properties: *position* (x/y coordinates), *color* (black, blue, yellow), *shape* (triangle, square, circle), and *size* (small, medium, large). The dataset also provides a structured representation of each image which we use in this paper. Figure 1 shows an example from the dataset.

Weakly supervised iterative search parser We use the semantic parser of Dasigi et al. (2019) which is a grammar-constrained encoder-decoder with attention model from Krishnamurthy et al. (2017). It learns to map a natural language utterance x into a program z such that it evaluates to the correct denotation $y = [\![z]\!]^r$ when executed against the structured image representation r. Dasigi et al. (2019) use a manually-designed, typed, variable-free, functional query language for NLVR, inspired by the GeoQuery language (Zelle and Mooney, 1996).

Given a dataset of triples (x_i, c_i, y_i), where x_i is an utterance, c_i is the set of images associated to it, and y_i is the set of corresponding denotations, their approach iteratively alternates between two phases to train the parser: Maximum marginal likelihood (MML) and a Reward-based method (RBM). In MML, for an utterance x_i, the model maximizes the marginal likelihood of programs in a given set of logical forms Z_i, all of which evaluate to the correct denotation. The set Z_i is constructed either by performing a heuristic search, or generated from a trained semantic parser.

The reward-based method maximizes the (approximate) expected value of a reward function $\mathcal{R}$.

$$\max_{\theta} \sum_{\forall i} \mathbb{E}_{\tilde{p}(z_i|x_i;\theta)} \mathcal{R}(x_i, z_i, c_i, y_i) \qquad (1)$$

Here, $\tilde{p}$ is the *re-normalization* of the probabilities assigned to the programs on the beam, and the reward function $\mathcal{R} = 1$ if z_i evaluates to the correct denotation for all images in c_i, or 0 otherwise. Please refer Dasigi et al. (2019) for details.

3 Consistency reward for programs

Consider the utterance x = *There is a yellow object above a black object* in Figure 1. There are many program candidates decoded in search that evaluate to the correct denotation. Most of them are *spurious*, i.e., they do not represent the meaning of the utterance and only coincidentally evaluate to the correct output. The semantic parser is expected to distinguish between the correct program and spurious ones by identifying correspondence between parts of the utterance and the program candidates. Consider a related utterance x' = *There are 2 boxes with a yellow object above a black object*. The parser should prefer programs for x and x' which contain similar sub-parts corresponding to the shared phrase p = *yellow object above a black object*. That is, the parser should be consistent in its interpretation of a phrase in different contexts. To incorporate this intuition during program search, we propose an additional reward to programs for an utterance that are consistent with programs for a related utterance.

Specifically, consider two related utterances x and x' that share a phrase p. We compute a reward for a program candidate z of x based on how similarly it maps the phrase p as compared to a program candidate z' of x'. To compute this reward we need (a) *relevant program parts* in z and z' that correspond to the phrase p, and (b) a *consistency reward* that measures consistency between those parts.

(a) Relevant program parts Let us first see how to identify relevant parts of a program z that correspond to a phrase p in the utterance.

Our semantic parser (from Krishnamurthy et al. (2017)) outputs a linearized version of the program $z = [z^1, \ldots, z^T]$, decoding one action at a time from the logical language. At each time step, the parser predicts a normalized attention vector over the tokens of the utterance, denoted by $[a_1^t, \ldots, a_N^t]$ for the z^t action. Here, $\sum_{i=1}^{N} a_i^t = 1$ and $a_i^t \geq 0$ for $i \in [1, N]$. We use these attention values as a relevance score between a program action and the utterance tokens. Given the phrase p with token span $[m, n]$, we identify the relevant actions in z as the ones whose total attention score over the tokens in p exceeds a heuristically-chosen threshold $\tau = 0.6$.

$$A(z, p) = \left\{ z^t \;\middle|\; t \in [1, T] \text{ and } \sum_{i=m}^{n} a_i^t \geq \tau \right\} \quad (2)$$

This set of program actions $A(z, p)$ is considered to be generated due to the phrase p. For example, for utterance *There is a yellow object above a black object*, with program objExists(yellow(above(black(allObjs)))), this approach could identify that for the phrase *yellow object above a black object* the actions corresponding to the functions yellow, above, and black are relevant.

(b) Consistency reward Now, we will define a reward for the program z based on how consistent its mapping of the phrase p is w.r.t. the program z' of a related utterance. Given a related program z' and its relevant action set $A(z', p)$, we define the consistency reward $S(z, z', p)$ as the F1 score for the action set $A(z, p)$ when compared to $A(z', p)$. If there are multiple shared phrases p_i between x and x', we can compute a weighted average of different $S(z, z', p_i)$ to compute a singular consistency reward $S(z, z')$ between the programs z and z'. In this work, we only consider a single shared phrase p between the related utterances, hence $S(z, z', p) = S(z, z', p)$ in our paper.

As we do not know the gold program for x', we decode top-K program candidates using beam-search and discard the ones that do not evaluate to the correct denotation. We denote this set of programs by Z'_c. Now, to compute a consistency reward $\mathcal{C}(x, z, x')$ for the program z of x,we take a weighted average of $S(z, z')$ for different $z' \in Z'_c$ where the weights correspond to the probability of the program z' as predicted by the parser.

$$\mathcal{C}(x, z, x') = \sum_{z' \in Z'_c} \tilde{p}(z'|x'; \theta) S(z, z') \quad (3)$$

Consistency reward based parser Given x and a related utterance x', we use $\mathcal{C}(x, z, x')$ as an additional reward in Eq. 1 to upweight programs for x that are consistent with programs for x'.

$$\max_{\theta} \sum_{\forall i} \mathbb{E}_{\tilde{p}(z_i|x_i;\theta)} \left[\mathcal{R}(x_i, z_i, c_i, y_i) + \mathcal{C}(x_i, z_i, x'_i) \right]$$

This consistency-based reward pushes the parser's probability mass towards programs that have consistent interpretations across related utterances, thus providing an additional training signal over simple denotation accuracy. The formulation presented in this paper assumes that there is a single related utterance x' for the utterance x. If multiple related utterances are considered, the consistency reward $\mathcal{C}(x, z, x'_j)$ for different related utterances x'_j can be summed/averaged to compute a single consistency reward $\mathcal{C}(x, z)$ the program z of utterance x based on all the related utterances.

4 Consistency in Language

The consistency reward (§3) makes a key assumption about the logical language in which the utterances are parsed: that the gold programs for utterances sharing a natural language phrase actually correspond to each other. For example, that the phrase *yellow object above a black object* would always get mapped to yellow(above(black)) irrespective of the utterance it occurs in.

On analyzing the logical language of Dasigi et al. (2019), we find that this assumption does not hold true. Let us look at the following examples:

x_1: *There are items of at least two different colors*
z_1: objColorCountGrtEq(2, allObjs)
x_2: *There is a box with items of at least two different colors*
z_2: boxExists(
memberColorCountGrtEq(2, allBoxes))

Here the phrase *items of at least two different colors*

Model	Dev		Test-P		Test-H	
	Acc.	Cons.	Acc.	Cons.	Acc.	Cons.
ABS. SUP. (Goldman et al., 2018)	84.3	66.3	81.7	60.1	-	-
ABS. SUP. + RERANK (Goldman et al., 2018)	85.7	67.4	84.0	65.0	82.5	63.9
ITERATIVE SEARCH (Dasigi et al., 2019)	85.4	64.8	82.4	61.3	82.9	64.3
+ Logical Language Design (ours)	88.2	73.6	86.0	69.6	-	-
+ Consistency Reward (ours)	**89.6**	**75.9**	**86.3**	**71.0**	**89.5**	**74.0**

Table 1: **Performance on NLVR:** Design changes in the logical language and consistency-based training, both significantly improve performance. Larger improvements in consistency indicate that our approach efficiently tackles spurious programs.

is interpreted differently in the two utterances. In x_2, a macro function memberColorCountGrtEq is used, which internally calls objColorCountGrtEq for each *box* in the image. Now consider,
x_3: *There is a tower with exactly one block*
z_3: boxExists(memberObjCountEq(1,allBoxes))
x_4: *There is a tower with a black item on the top*
z_4: objExists(black(top(allObjs)))
Here the phrase *There is a tower* is interpreted differently: z_3 uses a macro for filtering boxes based on their object count and interprets the phrase using boxExists. In the absence of a complex macro for checking *black item on the top*, z_4 resorts to using objExists making the interpretation of the phrase inconsistent. These examples highlight that these macros, while they shorten the search for programs, make the language inconsistent.

We make the following changes in the logical language to make it more consistent. Recall from §2 that each NLVR image contains 3 boxes each of which contains 1-8 objects. We remove macro functions like memberColorCountGrtEq, and introduce a generic boxFilter function. This function takes two arguments, a set of *boxes* and a filtering function f: Set[Obj] → bool, and prunes the input set of boxes to the ones whose objects satisfies the filter f. By doing so, our language is able to reuse the same object filtering functions across different utterances. In this new language, the gold program for the utterance x_2 would be
z_2: boxCountEq(1, boxFilter(allBoxes, objColorCountGrtEq(2)))
By doing so, our logical language can now consistently interpret the phrase *items of at least two different colors* using the object filtering function f: objColorCountGrtEq(2) across both x_1 and x_2. Similarly, the gold program for x_4 in the new logical language would be
z_4: boxExists(boxFilter(allBoxes, black(top)))
making the interpretation of *There is a box* consistent with x_3. Please refer appendix §A for details.

5 Experiments

Dataset We report results on the standard development, public-test, and hidden-test splits of NLVR. The training data contains 12.4k (utterance, image) pairs where each of 3163 utterances are paired with 4 images. Each evaluation set roughly contains 270 unique utterances.

Evaluation Metrics (1) *Accuracy* measures the proportion of examples for which the correct denotation is predicted. (2) Since each utterance in NLVR is paired with 4 images, a *consistency* metric is used, which measures the proportion of utterances for which the correct denotation is predicted for all associated images. Improvement in this metric is indicative of correct program prediction as it is unlikely for a spurious program to correctly make predictions on multiple images.

Experimental details We use the same parser, training methodology, and hyper-parameters as Dasigi et al. (2019). For discovering related utterances, we manually identify ∼10 sets of equivalent phrases that are common in NLVR. For example, *there are* NUM *boxes*, COLOR1 *block on a* COLOR2 *block*, etc. For each utterance that contains a particular phrase, we pair it with one other randomly chosen utterance that shares the phrase. We make 1579 utterance pairs in total. Refer appendix §B for details about data creation.[1]

Baselines We compare against the state-of-the-art models; ABS. SUP. (Goldman et al., 2018) that

[1]We release the data and code at https://www.github.com/nitishgupta/allennlp-semparse/tree/nlvr-v2/scripts/nlvr_v2

uses abstract examples, ABS. SUP. + RERANK that uses additional data and reranking, and the iterative search parser of Dasigi et al. (2019).

Results Table 1 compares the performance of our two proposed methods to enforce consistency in the decoded programs with the previous approaches. We see that changing the logical language to a more consistent one (§4) significantly improves performance: the accuracy improves by 2-4% and consistency by 4-8% on the dev. and public-test sets. Additionally, training the parser using our proposed consistency reward (§3) further improves performance: accuracy improves by 0.3-0.4% but the consistency significantly improves by 1.4-2.3%.[2] On the hidden-test set of NLVR, our final model improves accuracy by 7% and consistency by 10% compared to previous approaches. Larger improvements in consistency across evaluation sets indicates that our approach to enforce consistency between programs of related utterances greatly reduces the impact of spurious programs.

6 Conclusion

We proposed two approaches to mitigate the issue of spurious programs in weakly supervised semantic parsing by enforcing consistency between output programs. First, a consistency based reward that biases the program search towards programs that map the same phrase in related utterances to similar sub-parts. Such a reward provides an additional training signal to the model by leveraging related utterances. Second, we demonstrate the importance of logical language design such that it facilitates such consistency-based training. The two approaches combined together lead to significant improvements in the resulting semantic parser.

Acknowledgement

We would like to thank Pradeep Dasigi for helping us with the code for preprocessing NLVR and the Iterative Search model, Alane Suhr for getting us our model's evaluation results on the hidden test set in a timely manner, and the anonymous reviewers for their helpful comments. This work is supported in part by NSF award #IIS-1817183.

[2]We report average performance across 10 runs trained with different random seeds. All improvements in consistency are statistically significant (p-value < 0.05) based on the stochastic ordering test (Dror et al., 2019).

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from Question-Answer pairs. In *EMNLP*.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *CoNLL*.

Pradeep Dasigi, Matt Gardner, Shikhar Murty, Luke Zettlemoyer, and E. Hovy. 2019. Iterative search for weakly supervised semantic parsing. In *NAACL-HLT*.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance-how to properly compare deep neural models. In *ACL*.

Omer Goldman, Veronica Latcinnik, Udi Naveh, A. Globerson, and Jonathan Berant. 2018. Weakly-supervised semantic parsing with abstract examples. In *ACL*.

Kelvin Guu, Panupong Pasupat, E. Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. In *ACL*.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *EMNLP*.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V. Le, and N. Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *NeurIPS*.

Percy S. Liang, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. *Computational Linguistics*.

Alane Suhr, M. Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *ACL*.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*.

Luke S Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In *UAI '05*.

A Logical language details

In Figure 2, we show an example utterance with its gold program according to our proposed logical language. We use function composition and function currying to maintain the variable-free nature of our language. For example, action z^7 uses function composition to create a function from Set[Object] → bool by composing two functions, from Set[Object] → bool and Set[Object] → Set[Object]. Similarly, action z^{11} creates a function from Set[Object] → Set[Object] by composing two functions with the same signature.

Actions z^8 - z^{10} use function currying to curry the 2-argument function objectCountGtEq by giving it one int=2 argument. This results in a 1-argument function objectCountGtEq(2) from Set[Object] → bool.

B Dataset details

To discover related utterance pairs within the NLVR dataset, we manually identify 11 sets of phrases that commonly occur in NLVR and can be interpreted in the same manner:

1. { COLOR block at the base, the base is COLOR }
2. { COLOR block at the top, the top is COLOR }
3. { COLOR1 object above a COLOR2 object }
4. { COLOR1 block on a COLOR2 block, COLOR1 block over a COLOR2 block }
5. { a COLOR tower }
6. { there is one tower, there is only one tower, there is one box, there is only one box }
7. { there are exactly NUMBER towers, there are exactly NUMBER boxes }
8. { NUMBER different colors }
9. { with NUMBER COLOR items, with NUMBER COLOR blocks, with NUMBER COLOR objects }
10. { at least NUMBER COLOR items, at least NUMBER COLOR blocks, at least NUMBER COLOR objects }
11. { with NUMBER COLOR SHAPE, are NUMBER COLOR SHAPE, with only NUMBER COLOR SHAPE, are only NUMBER COLOR SHAPE }

In each phrase, we replace the abstract COLOR, NUMBER, SHAPE token with all possible options from the NLVR dataset to create grounded phrases. For example, *black block at the top*, *yellow object above a blue object*. For each set of equivalent grounded phrases, we identify the set of utterances that contains any of the phrase. For each utterance in that set, we pair it with 1 randomly chosen utterance from that set. Overall, we identify related utterances for 1420 utterances (out of 3163) and make 1579 pairings in total; if an utterance contains two phrases of interest, it can be paired with more than 1 utterance.

x: *There is one box with at least 2 yellow squares*
z: boxCountEq(1, boxFilter(allBoxes, objectCountGtEq(2)(yellow(square))))

Program actions for z:
z^1: bool → [<int,[Set[Box]:bool>, int, Set[Box]]
z^2: <int,[Set[Box]:bool> → boxCountEq
z^3: int → 1
z^4: Set[Box] → [<Set[Box],<Set[Object]:bool>:Set[Box]>, Set[Box], <Set[Object]:bool>]
z^5: <Set[Box],<Set[Object]:bool>:Set[Box]> → boxFilter
z^6: Set[Box] → allBoxes
z^7: <Set[Object]:bool> → [*, <Set[Object]:bool>, <Set[Object]:Set[Object]>]
z^8: <Set[Object]:bool> → [<int,Set[Object]:bool>, int]
z^9: <int,Set[Object]:bool> → objectCountGtEq
z^{10}: int → 2
z^{11}: <Set[Object]:Set[Object]> → [*, <Set[Object]:Set[Object]>, <Set[Object]:Set[Object]>]
z^{12}: <Set[Object]:Set[Object]> → yellow
z^{13}: <Set[Object]:Set[Object]> → square

Figure 2: Gold program actions for the utterance *There is one box with at least 2 yellow squares* according to our proposed logical language. The grammar-constrained decoder outputs a linearized abstract-syntax tree of the program in an in-order traversal.

An Improved Model for Voicing Silent Speech

David Gaddy and **Dan Klein**
University of California, Berkeley
{dgaddy,klein}@berkeley.edu

Abstract

In this paper, we present an improved model for voicing silent speech, where audio is synthesized from facial electromyography (EMG) signals. To give our model greater flexibility to learn its own input features, we directly use EMG signals as input in the place of hand-designed features used by prior work. Our model uses convolutional layers to extract features from the signals and Transformer layers to propagate information across longer distances. To provide better signal for learning, we also introduce an auxiliary task of predicting phoneme labels in addition to predicting speech audio features. On an open vocabulary intelligibility evaluation, our model improves the state of the art for this task by an absolute 25.8%.

1 Introduction

EMG-based voicing of silent speech is a task that aims to synthesize vocal audio from muscular signals captured by electrodes on the face while words are silently mouthed (Gaddy and Klein, 2020; Toth et al., 2009). While recent work has demonstrated a high intelligibility of generated audio when restricted to a narrow vocabulary (Gaddy and Klein, 2020), in a more challenging open vocabulary setting the intelligibility remained low (68% WER). In this work, we introduce an new model for voicing silent speech that greatly improves intelligibility.

We achieve our improvements by modifying several different components of the model. First, we improve the input representation. While prior work on EMG speech processing uses hand-designed features (Jou et al., 2006; Diener et al., 2015; Meltzner et al., 2018; Gaddy and Klein, 2020) which may throw away some information from the raw signals, our model learns directly from the complete signals with minimal pre-processing by using a set of convolutional neural network layers as feature extractors. This modification follows recent work in speech processing from raw waveforms (Collobert et al., 2016; Schneider et al., 2019) and gives our model the ability to learn its own features for EMG.

Second, we improve the neural architecture of the model. While other silent speech models have been based around recurrent layers such as LSTMs (Janke and Diener, 2017; Gaddy and Klein, 2020), we use the self-attention-based Transformer architecture (Vaswani et al., 2017), which has been shown to be a more powerful replacement across a range of tasks.

Finally, we improve the signal used for learning. Since the relatively small data sizes for this task creates a challenging learning problem, we introduce an auxiliary task of predicting phoneme labels to provide additional guidance. This auxiliary loss is inspired by prior work on the related problem of generating speech from ECoG sensors on the brain, which greatly benefited from intermediate prediction of phonemic information (Anumanchipalli et al., 2019).

We evaluate intelligibility of audio synthesized by our model on the single-speaker data from Gaddy and Klein (2020) in the most challenging open-vocabulary setting. Our results reflect an absolute improvement in error rate of 25.8% over the state of the art, from 68.0% to 42.2%, as measured by automatic transcription. Evaluation by human transcription gives an even lower error rate of 32%.

2 Model

At a high level, our system works by predicting a sequence of speech features from EMG signals and using a WaveNet vocoder (van den Oord et al., 2016) to synthesize audio from those predicted features, as was done in Gaddy and Klein (2020). The first component, dubbed the transduction model, takes in EMG signals from eight electrodes around

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 175–181
August 1–6, 2021. ©2021 Association for Computational Linguistics

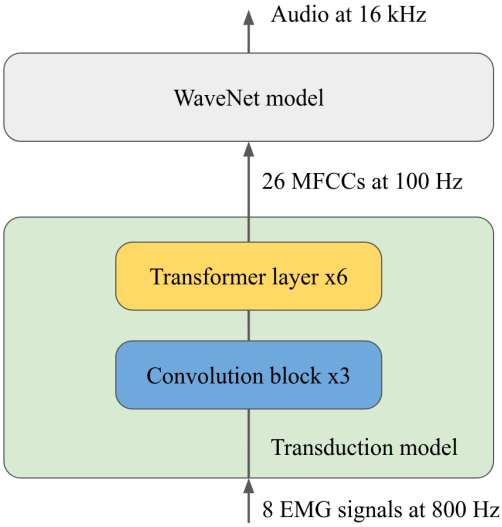


Figure 1: Model overview

the face and outputs a sequence of speech features represented as Mel-frequency cepstral coefficients (MFCCs). The final step of vocoding audio from MFCCs is unchanged in our work, so we defer to Gaddy and Klein (2020) for the details of the WaveNet model.

The neural architecture for our transduction model is made up of a set of residual convolution blocks followed by a transformer with relative position embeddings, as shown in Figure 1. We describe these two components in Sections 2.1 and 2.2 below. Next, in Section 2.3 we describe our training procedure, which aligns each silent utterance to a corresponding vocalized utterance as in Gaddy and Klein (2020) but with some minor modifications. Finally, in Section 2.4 we describe the auxiliary phoneme-prediction loss that provides additional signal to our model during training.[1]

2.1 Convolutional EMG Feature Extraction

The convolutional layers of our model are designed to directly take in EMG signals with minimal preprocessing. Prior to use of the input EMG signals, AC electrical noise is removed using band stop filters at harmonics of 60 Hz, and DC offset and drift are removed with a 2 Hz high-pass filter. The signals are then resampled from 1000 Hz to 800 Hz, and the magnitudes are scaled down by a factor of 10.

Our convolutional architecture uses a stack of 3 residual convolution blocks inspired by ResNet (He et al., 2016), but modified to use 1-dimensional

[1]Code for our model is available at https://github.com/dgaddy/silent_speech.

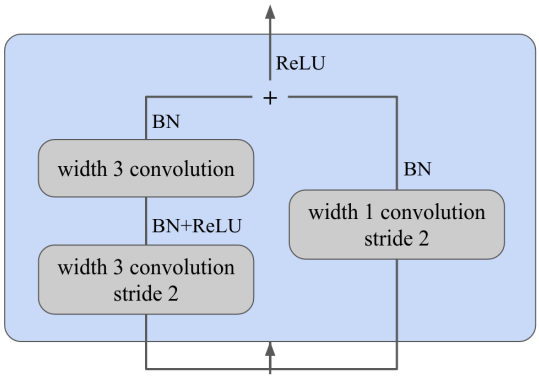


Figure 2: Convolution block architecture

convolutions. The architecture used for each convolution block is shown in Figure 2, and has two convolution-over-time layers along the main path as well as a shortcut path that does not do any aggregation over the time dimension. Each block downsamples the signal by a factor of 2, so that the input signals at 800 Hz are eventually transformed into features at 100Hz to match the target speech feature frame rate. All convolutions have channel dimension 768.

Before passing the convolution layer outputs to the rest of the model, we include an embedding of the session index, which helps the model account for differences in electrode placement after electrodes are reattached for each session. Each session is represented with a 32 dimensional embedding, which is projected up to 768 dimensions with a linear layer before adding to the convolution layer outputs at each timestep.

2.2 Transformer with Relative Position Embeddings

To allow information to flow across longer time horizons, we use a set of bidirectional Transformer encoder layers (Vaswani et al., 2017) on top of the convolution layers in our model. To capture the time-invariant nature of the task, we use relative position embeddings as described by Shaw et al. (2018) rather than absolute position embeddings. In this variant, a learned vector p that depends on the relative distance between the query and key positions is added to the key vectors when computing attention weights. Thus, the attention logits are computed with

$$e_{ij} = \frac{(W_K x_j + p_{ij})^\top (W_Q x_i)}{\sqrt{d}}$$

where p_{ij} is an embedding lookup with index $i - j$, up to a maximum distance k in each direction (x are inputs to the attention module, W_Q and W_K are query and key transformations, and d is the dimension of the projected vectors $W_Q x_i$). For our model, we use $k = 100$ (giving each layer a 1 second view in each direction) and set all attention weights with distance greater than k to zero. We use six of these Transformer layers, with 8 heads, model dimension 768, feedforward dimension 3072, and dropout 0.1.

The output of the last Transformer layer is passed through a final linear projection down to 26 dimensions to give the MFCC audio feature predictions output by the model.

2.3 Alignment and Training

Since silent EMG signals and vocalized audio features must be recorded separately and so are not time-aligned, we must form an alignment between the two recordings to calculate a loss on predictions from silent EMG. Our alignment procedure is similar to the predicted-audio loss used in Gaddy and Klein (2020), but with some minor aspects improved.

Our loss calculation takes in a sequence of MFCC features $\hat{A}_S$ predicted from silent EMG and another sequence of target features A_V from a recording of vocalized audio for the same utterance. We compute a pairwise distance between all pairs of features

$$\delta[i, j] = \left\| A_V[i] - \hat{A}_S[j] \right\|_2$$

and run dynamic time warping (Rabiner and Juang, 1993) to find a minimum-cost monotonic alignment path through the δ matrix. We represent the alignment as $a[i] \rightarrow j$ with a single position j in $\hat{A}_S$ for every index i in A_V, and take the first such position when multiple are given by dynamic time warping. The loss is then the mean of aligned pairwise distances:

$$L = \frac{1}{N_V} \sum_{i=1}^{N_V} \delta[i, a[i]]$$

In addition to the silent-EMG training, we also make use of EMG recordings during vocalized speech which are included in the data from Gaddy and Klein (2020). Since the EMG and audio targets are recorded simultaneously for these vocalized examples, we can calculate the pairwise distance loss directly without any dynamic time warping. We train on the two speaking modes simultaneously.

To perform batching across sequences of different lengths during training, we concatenate a batch of EMG signals across time then reshape to a batch of fixed-length sequences before feeding into the network. Thus if the fixed batch-sequence-length is l, the sum of sample lengths across the batch is N_S, and the signal has c channels, we reshape the inputs to size $(\lceil N_S / l \rceil, l, c)$ after zero-padding the concatenated signal to a multiple of l. After running the network to get predicted audio features, we do the reverse of this process to get a set of variable-length sequences to feed into the alignment and loss described above. This batching strategy allows us to make efficient use of compute resources and also acts as a form of dropout regularization where slicing removes parts of the nearby input sequence. We use a sequence length $l = 1600$ (2 seconds) and select batches dynamically up to a total length of $N_{Smax} = 204800$ samples (256 seconds).

We train our model for 80 epochs using the AdamW optimizer (Loshchilov and Hutter, 2017). The peak learning rate is 10^{-3} with a linear warm-up of 500 batches, and the learning rate is decayed by half after 5 consecutive epochs of no improvement in validation loss. Weight decay 10^{-7} is used for regularization.

2.4 Auxiliary Phoneme Loss

To provide our model with additional training signal and regularize our learned representations, we introduce an auxiliary loss of predicting phoneme labels at each output frame.

To get phoneme labels for each feature frame of the vocalized audio, we use the Montreal Forced Aligner (McAuliffe et al., 2017). The aligner uses an acoustic model trained on the LibriSpeech dataset in conjunction with a phonemic dictionary to get time-aligned phoneme labels from audio and a transcription.

We add an additional linear prediction layer and softmax on top of the Transformer encoder to predict a distribution over phonemes. For training, we modify the alignment and loss cost δ by appending a term for phoneme negative log likelihood:

$$\delta'[i, j] = \left\| A_V[i] - \hat{A}_S[j] \right\|_2 - \lambda P_V[i]^\top \log \hat{P}_S[j]$$

where $\hat{P}_S$ is the predicted distribution from the model softmax and P_V is a one-hot vector for the target phoneme label. We use $\lambda = .1$ for the phoneme loss weight. After training, the phoneme prediction layer is discarded.

Model	WER
Gaddy and Klein (2020)	68.0
This work	**42.2**
Ablation: Replace convolution features with hand-designed features	45.2
Ablation: Replace Transformer with LSTM	46.0
Ablation: Remove phoneme loss	51.7

Table 1: Open vocabulary word error rate results from an automatic intelligibility evaluation.

3 Results

We train our model on the open-vocabulary data from Gaddy and Klein (2020). This data contains 19 hours of facial EMG data recordings from a single English speaker during silent and vocalized speech. Our primary evaluation uses the automatic metric from that work, which transcribes outputs with an automatic speech recognizer[2] and compares to a reference with a word error rate (WER) metric. We also evaluate human intelligibility in Section 3.1 below.[3]

The results of the automatic evaluation are shown in Table 1. Overall, we see that our model improves intelligibility over prior work by an absolute 25.8%, or 38% relative error reduction. Also shown in the table are ablations of our three primary contributions. We ablate the convolutional feature extraction by replacing those layers with the hand-designed features used in Gaddy and Klein (2020), and we ablate the Transformer layers by replacing with LSTM layers in the same configuration as that work (3 bidirectional layers, 1024 dimensions). To ablate the phoneme loss, we simply set its weight in the overall loss to zero. All three of these ablations show an impact on our model's results.

3.1 Human Evaluation

In addition to the automatic evaluation, we performed a human intelligibility evaluation using a similar transcription test. Two human evaluators without prior knowledge of the text were asked to listen to 40 synthesized samples and write down the words they heard (see Appendix A for full instructions given to evaluators). We then compared these transcriptions to the ground-truth reference with a WER metric.

The resulting word error rates from the two human evaluators' transcriptions are 36.1% and 28.5% (average: 32.3%), compared to 42.2% from automatic transcriptions. These results validate the improvement shown in the automatic metric, and indicate that the automatic metric may be underestimating intelligibility to humans. However, the large variance across evaluators shows that the automatic metric may still be more appropriate for establishing consistent evaluations across different work on this task.

4 Phoneme Error Analysis

One additional advantage to using an auxiliary phoneme prediction task is that it provides a more easily interpretable view of model predictions. Although the phoneme predictions are not directly part of the audio synthesis process, we have observed that mistakes in audio and phoneme prediction are often correlated. Therefore, to better understand the errors that our model makes, we analyze the errors of our model's phoneme predictions. To analyze the phoneme predictions, we align predictions on a silent utterance to phoneme labels of a vocalized utterance using the procedure described above in Sections 2.3 and 2.4, then evaluate the phonemes using the measures described in Sections 4.1 and 4.2 below.

4.1 Confusion

First, we measure the confusion between each pair of phonemes. We use a frequency-normalized metric for confusion: $(e_{p1,p2} + e_{p2,p1})/(f_{p1} + f_{p2})$, where $e_{p1,p2}$ is the number of times $p2$ was predicted when the label was $p1$, and f_{p1} is the number of times phoneme $p1$ appears as a target label. Figure 3 illustrates this measure of confusion using darkness of lines between the phonemes, and Appendix B lists the values of the most confused pairs.

We observe that many of the confusions are be-

[2] An implementation of DeepSpeech (Hannun et al., 2014) from Mozilla (https://github.com/mozilla/DeepSpeech)

[3] Output audio samples available at https://dgaddy.github.io/silent_speech_samples/ACL2021/.

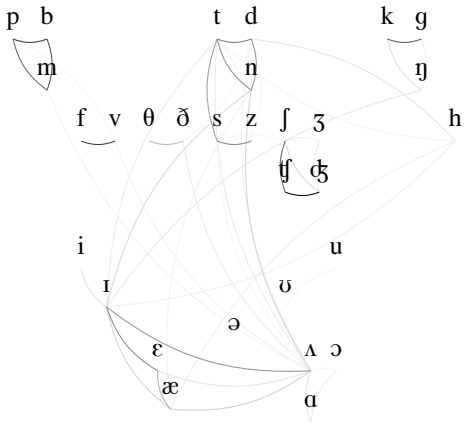


Figure 3: Phoneme confusability (darker lines indicate more confusion - maximum darkness is 13% confusion)

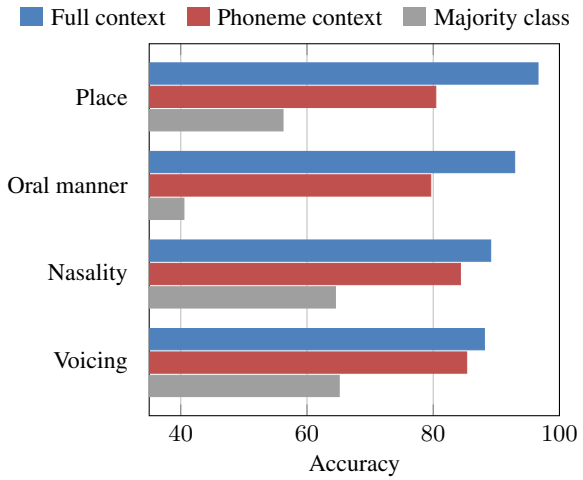


Figure 4: Accuracy of selecting phonemes along articulatory feature dimensions. We compare our full EMG model (full context) with a majority class baseline and a model given only phoneme context as input.

tween pairs of consonants that differ only in voicing, which is consistent with the observation in Gaddy and Klein (2020) that voicing signals appear to be subdued in silent speech. Another finding is a confusion between nasals and stops, which is challenging due to the role of the velum and its relatively large distance from the surface electrodes, as has been noted in prior work (Freitas et al., 2014). We also see some confusion between vowel pairs and between vowels and consonants, though these patterns tend to be less interpretable.

4.2 Articulatory Feature Accuracy

To better understand our model's accuracy across different consonant articulatory features, we perform an additional analysis of phoneme selection across specific feature dimensions. For this analysis, we define a confusion set for an articulatory feature as a set of English phonemes that are identical across all other features. For example, one of the confusion sets for the place feature is $\{p, t, k\}$, since these phonemes differ in place of articulation but are the same along other axes like manner and voicing (a full listing of confusion sets can be found in Appendix C). For each feature of interest, we calculate a forced-choice accuracy within the confusion sets for that feature. More specifically, we find all time steps in the target sequence with labels belonging in a confusion set and restrict our model output to be within the corresponding set for those positions. We then compute an accuracy across all those positions that have a confusion set.

To evaluate how much of the articulatory feature accuracies can be attributed to contextual inferences rather than information extracted from EMG, we compare our results to a baseline model that is trained to make decisions for a feature based on nearby phonemes. In the place of EMG feature inputs, this baseline model is given the sequence of phonemes predicted by the full model, but with information about the specific feature being tested removed by collapsing phonemes in each of its confusion sets to a single symbol. Additional details on this baseline model can be found in Appendix C.

The results of this analysis are shown in Figure 4. By comparing the gap in accuracy between the full model and the phoneme context baseline, we again observe trends that correspond to our prior expectations. While place and oral manner features can be predicted much better by our EMG model than from phonemic context alone, nasality and voicing are more challenging and have a smaller improvement over the contextual baseline.

5 Conclusion

By improving several model components for voicing silent speech, our work has achieved a 38% relative error reduction on this task. Although the problem is still far from solved, we believe the large rate of improvement is a promising sign for continued progress.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1618460 and by DARPA under the LwLL program / Grant No. FA8750-19-1-0504.

References

Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. 2019. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

Lorenz Diener, Matthias Janke, and Tanja Schultz. 2015. Direct conversion from facial myoelectric signals to speech using deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

João Freitas, António JS Teixeira, Samuel S Silva, Catarina Oliveira, and Miguel Sales Dias. 2014. Velum movement detection based on surface electromyography for speech interface. In *BIOSIGNALS*, pages 13–20.

David Gaddy and Dan Klein. 2020. Digital voicing of silent speech. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5521–5530, Online. Association for Computational Linguistics.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Matthias Janke and Lorenz Diener. 2017. EMG-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2375–2385.

Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. 2006. Towards continuous speech recognition using surface electromyography. In *Ninth International Conference on Spoken Language Processing*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering*, 15(4):046031.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *ArXiv*, abs/1609.03499.

Lawrence Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of speech recognition*. Prentice Hall.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *INTERSPEECH*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Arthur R. Toth, Michael Wand, and Tanja Schultz. 2009. Synthesizing speech from electromyography using voice transformation techniques. In *INTERSPEECH*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A Instructions to Human Evaluators

The following instructions were given to human evaluators for the transcription test described in Section 3.1:

Please listen to each of the attached sound files and write down what you hear. There are 40 files, each of which will contain a sentence in English. Write your transcriptions into a spreadsheet such as Excel or Google sheets so that the row numbers match the numbers in the file names. Many of the clips may be difficult to hear. If this is the case, write whatever words you are able to make out, even if it does not form a complete expression. If you are not entirely sure about a word but can make a strong guess, you may include it in your transcription, but only do so if you beleive it is more likely than not to be the correct word. If you cannot make out any words, leave the corresponding row blank.

B Phoneme Confusability

This section provides numerical results for phoneme confusions to complement the illustration given in Section 4.1 of the main paper. We compare the frequency of errors between two phonemes to the frequency of correct predictions on those phonemes. We define the following two quantities:

$$\text{Confusion: } (e_{p1,p2} + e_{p2,p1})/(f_{p1} + f_{p2})$$

$$\text{Accuracy: } (e_{p1,p1} + e_{p2,p2})/(f_{p1} + f_{p2})$$

where $e_{p1,p2}$ is the number of times $p2$ was predicted when the label was $p1$, and f_{p1} is the number of times phoneme $p1$ appears as a target label. Results for the most confused pairs are shown in the table below.

Phonemes		Confusion (%)	Accuracy (%)
ʤ	ʧ	13.2	49.4
v	f	10.4	72.0
p	b	10.3	64.3
m	b	9.3	74.3
k	ɡ	8.9	77.2
ʃ	ʧ	8.3	59.8
p	m	8.1	73.0
t	d	7.2	64.0
z	s	6.6	80.0
ɪ	ɛ	6.5	60.6
t	n	6.3	67.1
n	d	6.0	66.8
ɪ	ʌ	6.0	65.8
ɹ	ɚ	5.7	78.2
t	s	5.5	72.8
ɛ	æ	4.7	70.9
u	oʊ	4.3	77.4
θ	ð	4.1	76.9
ʌ	æ	3.2	72.1
ɪ	æ	3.1	64.9

C Articulatory Feature Analysis Details

The following table lists all confusion sets used in our articulatory feature analysis in Section 4.2.

Feature	Confusion Sets
Place	{p,t,k} {b,d,g} {m,n,ŋ} {f,θ,s,ʃ,h} {v,ð,z,ʒ}
Oral manner	{t,s} {d,z,l,r} {ʃ,ʧ} {ʒ,ʤ}
Nasality	{b,m} {d,n} {g,ŋ}
Voicing	{p,b} {t,d} {k,g} {f,v} {θ,ð} {s,z} {ʃ,ʒ} {ʧ,ʤ}

The phoneme context baseline model uses a Transformer architecture with dimensions identical to our primary EMG-based model, but is fed phoneme embeddings of dimension 768 in the place of the convolutional EMG features. The phonemes input to this model are the maximum-probability predictions output by our primary model at each frame, but with all phonemes from a confusion set replaced with the same symbol. We train a separate baseline model for each of the four articulatory feature types to account for different collapsed sets in the input. During training, a phoneme likelihood loss is applied to all positions and no restrictions are enforced on the output. Other training hyperparameters are the same between this baseline and the main model.

D Additional Reproducability Information

All experiments were run on a single Quadro RTX 6000 GPU, and each took approximately 12 hours. Hyperparameters were tuned manually based on automatic transcription WER on the validation set. The phoneme loss weight hyperparameter λ was chosen from $\{1, .5, .1, .05, .01, .005\}$. We report numbers on the same test split as Gaddy and Klein (2020), but increase the size of the validation set to 200 examples to decrease variance during model exploration and tuning. Our model contains approximately 40 million parameters.

What's in the Box? A Preliminary Analysis of Undesirable Content in the Common Crawl Corpus

Alexandra (Sasha) Luccioni
Université de Montréal &
Mila Québec AI Institute
sasha.luccioni@mila.quebec

Joseph D. Viviano
Mila Québec AI Institute
joseph@viviano.ca

Abstract

Whereas much of the success of the current generation of neural language models has been driven by increasingly large training corpora, relatively little research has been dedicated to analyzing these massive sources of textual data. In this exploratory analysis, we delve deeper into the Common Crawl, a colossal web corpus that is extensively used for training language models. We find that it contains a significant amount of undesirable content, including hate speech and sexually explicit content, even after filtering procedures. We discuss the potential impacts of this content on language models and conclude with future research directions and a more mindful approach to corpus collection and analysis.

1 Introduction

In recent years, much of the progress in Natural Language Processing (NLP) research has been largely driven by Transformer-based language models, which have pushed forward the state-of-the-art in tasks such as question answering (Rajpurkar et al., 2018) and natural language inference (Bowman et al., 2015). However, these increasingly complex models also require increasingly large amounts of data to train them, which is often a combination of curated, high-quality datasets such as encyclopedic articles and books and non-curated content from the Web (Radford et al., 2018, 2019). This second category of large, non-curated dataset is becoming increasingly popular as they are required to train large language models.

The current largest dataset used for training neural language models, the Common Crawl, is a non-curated corpus consisting of multilingual snapshots of the web. New versions of the Common Crawl are released monthly, with each version containing 200 to 300 TB of textual content scraped via automatic web crawling. This dwarfs other commonly used corpora such as English-language Wikipedia, which adds up to roughly 5.6 TB of data, and the BookCorpus, which only represents around 6 GB (Zhu et al., 2015). The Common Crawl has been used to train many of the recent neural language models in recent years, including the GPT model series (Radford et al., 2018; Brown et al., 2020), BERT (Devlin et al., 2018) and FastText (Grave et al., 2018) and, given its size, often represents the majority of data used to train these architectures.

In the current article, we present an initial analysis of the Common Crawl, highlighting the presence of several types of explicit and abusive content even after filtering. We discuss our findings and, given the potential downstream impact of this content on language models, we discuss the importance of ensuring that the corpora we use for training language models are extracted more mindfully and with more emphasis on their quality and propose avenues of research to achieve this goal.

2 Related Work

In recent years, a growing body of research in NLP has unearthed biases in common language models (Bolukbasi et al., 2016; Sheng et al., 2019; Zhao et al., 2019; Bordia and Bowman, 2019; Hutchinson et al., 2020). This work has raised important questions regarding the impact of these embedded biases on downstream decision-making, given the increasing usage of these models in various applications. Consequently, much work has also been dedicated to creating standardized diagnostic tests to detect these biases (Caliskan et al., 2017; May et al., 2019; Nadeem et al., 2020; Sweeney and Najafian, 2019) and to remove them (Bolukbasi et al., 2016; Zhao et al., 2018; Manzini et al., 2019), although the extent to which this is possible is still under debate (Gonen and Goldberg, 2019). In fact, research has found that *"The biases found in Internet-scale language models like GPT-2 are representative of the data on which the model was trained"* (So-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 182–189
August 1–6, 2021. ©2021 Association for Computational Linguistics

laiman et al., 2019), which can be directly linked to the presence of hate speech on the Internet (Abid et al., 2021).

However, given the importance of this research, comparatively little attention has been dedicated to analyzing the corpora used to train language models. This is understandable because frequently used datasets such as the Common Crawl contain truly massive amounts of data, making it challenging to mine it for meaningful insights. In fact, a recent survey on automatic web page classification has deemed the task difficult not only due to the complexity and heterogeneity of web content, but also due its the high computational cost, suggesting that machine learning (ML) approaches have much to contribute to it (Hashemi, 2020). While certain notable endeavors have indeed analyzed specific aspects of corpora such as the Common Crawl (Kolias et al., 2014; Caswell et al., 2021) and Wikipedia (Hube, 2017), they have only scratched the surface of what these bodies of text contain. For instance, recent work has found that the Common Crawl contained over 300,000 documents from unreliable news sites and banned subReddit pages containing hate speech and racism (Gehman et al., 2020), while complementary research has shown that individual training examples can be extracted by querying language models (Carlini et al., 2020), together illustrating that the presence of questionable content is a significant issue for statistical language models. In the current work, we endeavor to understand the content and quality of the Common Crawl as a first step towards establishing more consistent approaches to filtering and refining it.

3 Analyzing the Common Crawl

Given its size, both downloading and analyzing the Common Crawl are time-consuming and costly endeavors. The most recent version of the Common Crawl, dating from November/December 2020, has 2.6 billion web pages in raw text format, saved in 'shards' each containing of tens of thousands of pages. Given our hardware constraints, we chose to focus on a subset of the corpus, randomly sampling 1% of the files it contains, which after filtering by language amounts to roughly 115 GB of textual content or 5,835,339 web pages in total, which we analyzed in terms of hate speech, adult content, and efficacy of perplexity-based filtering [1]. In this work, we focus on detecting sexually-explicit and hate speech, since they represent common examples of "undesirable" content that can be generally seen as inappropriate for a language model to generate in most situations. We acknowledge that desirable model behaviour is application specific, and believe our findings can extend to any other "undesirable" topic that might be present in available language corpora. We present our results in the sections below.

3.1 Detecting Hate Speech

The existence of hate speech on the internet has been described as "an important societal problem of our time", with "profound and lasting" psychological effects on its victims (Mishra et al., 2019). As such, a substantial amount of NLP research dedicated to automating hate speech detection, with several datasets and approaches being proposed in recent years (Schmidt and Wiegand, 2017; Mishra et al., 2019; Vidgen and Derczynski, 2020; Kiritchenko and Mohammad, 2018). Most of this research is carried out on data extracted from social media sources such as Twitter (Founta et al., 2018; Basile et al., 2019; Waseem and Hovy, 2016) and Reddit (Tadesse et al., 2019; Farrell et al., 2019), with both ML-based (Badjatiya et al., 2017) and count-based approaches (Davidson et al., 2017) achieving comparable results (Fortuna and Nunes, 2018). In order to estimate the quantity of hate speech in the Common Crawl, we endeavored to compare 3 approaches: DELIMIT, a recent BERT-based model trained on social media data (Aluru et al., 2020), Hate Sonar, a Logistic Regression approach trained on data from Web fora and Twitter (Davidson et al., 2017) and a n-gram-based approach using a list of n-grams extracted from Hate Base. We present samples of text flagged by all of these approaches in Table 1, below.

We found that the three approaches compared suggest similar proportions of websites containing hate speech : 5.24% of websites from our sample were flagged by DELIMIT, 4.02% by HateSonar, and 6.38% by the n-gram approach [2]. Qualitative analysis of a sample of sites flagged by each approach showed that while n-grams picked up on racial slurs, HateSonar also detected debates about racial supremacy and racially-charged conspiracy theories. Many of the sites that DELIMIT

[1] All code used in these analysis are publicly available: https://github.com/josephdviviano/whatsinthebox

[2] We are conscious of the high false positive rate of n-gram approaches and therefore only consider sites to be flagged if they contain 3 or more n-grams from the list.

Approach	Text
HateSonar	Their US/Euro plan put in your face: demonic jews hate white goyim!
	Such sick and twisted people, white people are.
Delimit	they are only stupid arab from wp-ar haha
	Yeah, dumb ass n*gger †
N-gram	nude attention whore asian bastards
	In America all male look like this homo

Table 1: Examples of hate speech found by the approaches tested. Examples with † have been censored by the authors.

Page URL (`http://` removed)
`adultmovietop100.com/`
`erohon.me/`
`celebrityfan.net/`
`queantube.com/`
`adelaide-femaleescorts.webcam`

Table 2: Sample of URLs of adult content websites identified by the n-gram approach. Protocol removed to prevent URL generation.

flagged were adult content with mentions of violent acts towards specific ethnic groups, illustrating the fine line between sexual violence and hate speech, which we elaborate further in the following subsection. Generally speaking, the presence of even a small fraction of websites that incite hate in training corpora is worrisome since it can result in models that replicate this kind of discourse when prompted (Wolf et al., 2017; Carlini et al., 2020).

3.2 Sexually Explicit Content

Compared to hate speech, the detection of sexually explicit content has received less attention from the NLP community, with existing ML approaches focusing mainly on the detection of explicit images (Wehrmann et al., 2018; Rowley et al., 2006) and URLs (Matic et al., 2020), whereas n-gram-based approaches remain predominantly used in practice by web providers (Hammami et al., 2003; Polpinij et al., 2006; Ho and Watters, 2004). In our analysis, we used a list of n-grams extracted from adult websites in order to establish the percentage of websites from our sample that contained sexually explicit content; however, we found no available statistical or ML-based approach that we could use to compare our count-based approach with. The n-gram approach detected that 2.36% of the web pages that we analyzed contained at least one of the words from our list, with 1.36% containing 3 or more and 0.73% containing 10 or more (see Table 3 for results). We show a sample of the URLs flagged by our approach in Table 2, below.

While a few percent of sexually explicit content may not seem like much, the type of language and content contained on adult websites can have harmful repercussions. For instance, the prevalence of sexual violence towards women, especially towards women of color, on adult websites (Foubert et al., 2019; Shim et al., 2015; Fritz et al., 2020) may contribute to further dissemination and amplification of these biases in downstream models. As modern language models have no way to evaluate generation appropriateness, models trained with even a small proportion of these undesirable inputs cannot be guaranteed to avoid generating outputs with similar biases if presented with a specific context or prompt. This is a risk that is important to mitigate in applications, where the general-purpose language models can end up being used in applications used by sensitive groups in professional contexts or minors, such as chatbots and toys.

3.3 Filtering by Perplexity Score

While the analyses described above were carried out on unfiltered web pages from the Common Crawl, the training pipeline of many large-scale NLP models involves some type of filtering and cleaning, from excluding low-quality content (Grave et al., 2018) to fuzzy deduplication (Brown et al., 2020). One such popular filtering approach is based on training a language model on a target, high-quality domain such as Wikipedia, and using it to calculate the perplexity score of web pages using this model (Wenzek et al., 2020). To test the efficacy of this scoring procedure, we calculated the perplexity score of each web page from our sample of the Common Crawl and used it to separate pages into 3 equal buckets (high, middle and low-quality) based on their perplexity. We compare the percentages of hate speech and sexually explicit content for the entire sample, as well as the high- and low-quality documents, in Table 3.

While filtering by perplexity does seem to filter out many websites containing sexual content, it does not detect much of the hate speech that is flagged by the count-based or statistical methods. In fact, perplexity scores had low correlations with all detection methods tested (Figure 1). This supports the methodology of Wenzek et al. (2020),

	Entire Sample	High Quality	Low Quality
1+ sexual n-grams	2.36%	1.81%	3.97%
3+ sexual n-grams	1.36%	0.42%	3.11%
10+ sexual n-grams	0.73%	0.08%	1.98%
1+ hate n-grams	17.78%	18.95%	17.19%
3+hate n-grams	6.38%	6.19%	8.26%
10+ hate n-grams	1.16%	1.17%	1.70%
Hate speech (Sonar)	4.02%	3.47%	5.09%
Hate speech (Delimit)	5.24%	5.77%	5.66%

Table 3: Comparison of hate speech and sexual content detected in the entire corpus, as well as high- and low-quality sites.

who noted that while *"perplexity was a relative good proxy for quality"*, also argued that some of the lower-quality texts could still be useful for specific applications, and therefore did not use it to exclude documents from the training set of their language model. While we are exploring ways of modifying the original approach in order to be more discerning, we believe that there more nuanced metrics that can be used for estimating and filtering documents based on text, potentially coupling embedding-based approaches with statistical ones.

3.4 Behaviour of Different Detection Methods

The approaches that we compared in the current study are different in the features that they use and techniques employed for detecting particular types of content. HateSonar employs classical NLP techniques for hate speech detection, constructing features from Penn Part-of-Speech N-grams with TF-IDF weighting based on a hand-crafted hate speech dataset, training simple classifier ensembles using Support Vector Machines, random forests, naive Bayes, and linear models. Delimit, on the other hand, is A BERT-based model trained on Twitter and Reddit posts, not relying on any handcrafted features. Our simple n-gram approach unsurprisingly was more in agreement with HateSonar than Delimit, given that both rely on count-based features. The fact that all methods identified different instances of clear hate speech implies that we are far from a general purpose dataset-filtering approach. These results also imply that deep learning models learn very different features to classify hate speech than other methods, and given their sensitivity to the specific composition of the dataset used to train them (as exposed by the propensity of large models to memorize training examples (Carlini et al., 2020)), the presence of undesirable content in the corpora used to train them should be taken seriously.

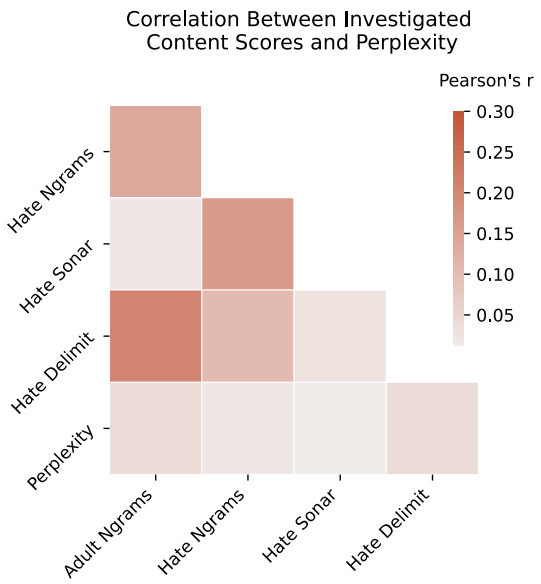


Figure 1: Correlation coefficients (Pearson's r) calculated between all content metrics investigated and perplexity, a commonly-used text quality metric.

4 Discussion

4.1 Summary of Results

We recognize that the exploratory work presented above is only the tip of the iceberg in terms of the analyses that can be done on the massive web corpora that are feeding our language models. However, analyzing the Common Crawl would require computational resources far in excess of what is available to most research institutions. We therefore hope that this initial analysis will inspire our fellow researchers to continue to dig deeper into this topic, and to propose more scalable, thorough, and nuanced approaches for analyzing the massive corpora used to train language models. We also recognize this analysis would have been more comprehensive on a small curated dataset, but given the

amount of data needed to train modern language models, we believe the community needs to move beyond analysis techniques only compatible with small-data, toward something that will scale to the datasets used to train these large models.

Also, while we have currently adopted a purely descriptive approach, we feel that it is worth discussing and debating the consequences of our analysis, and those of our peers, within the NLP community. While it can be argued that the Common Crawl corpus is an accurate portrayal of the discourse of modern society – which includes sexual content, hate speech, and racial and gender biases – we believe that it is up for debate whether this discourse is the one that we, as a community, want to use to train the models that translate our texts, influence our search results and answer our questions. Notably, the Common Crawl over-represents those populations that are avid users of the internet: younger, English-speaking individuals from developed countries, who are those who have the most access to the internet globally (World Bank, 2018). Furthermore, internet communities supported by anonymity and and particular norms can amplify toxic discourse that would not be found in mainstream corpora (Massanari, 2017) often exacerbated by the well-documented 'online disinhibition' phenomenon where users find themselves more likely to engage in anti-social behaviours due to the lack of immediate social feedback (Wachs et al., 2019; Mathew et al., 2019; de Lima et al., 2021). This can further perpetuate the lack of diverse, representative language models that can adequately mirror society beyond the boundaries of internet communities.

4.2 Future Work

Given the general superior performance of large language models on common benchmarks, and that they require ever larger datasets to train them, we believe it is important that for the ML community to carry out a more extensive analysis of: 1) the impact of undesirable content in the datasets used to train these models on downstream performance; 2) the effect of properly filtering these examples out of the dataset *before* model training, and 3) approaches for regularizing model outputs to be acceptable regardless of the data used to train the model. All three directions require a better understanding of the contents of the datasets, which we believe requires new tools that are scalable to the Common Crawl (or similarly large and diverse corpora) to identify such examples. Models trained to detect undesirable examples, like the ones used in this paper, need to be improved such that they can reliably generalize to the Common Crawl, which constitutes a significant undertaking. Additionally, future work could explore the utility of controlling model generation using labelled "undesirable" examples (Zhang et al., 2020; Engel et al., 2017), or human-in-the-loop learning methods (Wang et al., 2021) for fine-tuning a language model trained using undesirable examples. It will also be important to evaluate whether curation is sufficient: it remains possible that a model could create an undesirable generation from multiple distinct innocuous examples (Bender et al., 2021; Gehman et al., 2020). It is also worth considering that for some applications, task-focused models with curated training examples may perform better than large models trained on unfiltered corpora, so that their behaviour can be more reliably guaranteed: these are all interesting avenues for future work.

Finally, while larger corpora generally result in better models (Kaplan et al., 2020; Sun et al., 2017), data quality and corpora content also plays a major role in the caliber and appropriateness of these models for the various downstream applications (Florez, 2019; Abid et al., 2021; Bhardwaj et al., 2021). To produce high quality and safe neural language models will likely require the community to adopt more mindful data collection practices (Gehman et al., 2020; Bender and Friedman, 2018; Gebru et al., 2018; Jo and Gebru, 2020; Paullada et al., 2020; Bender et al., 2021), establish standardized filtering pipelines for corpora (Roziewski and Stokowiec, 2016; Ortiz Suarez et al., 2019; Wenzek et al., 2020), and develop methods for evaluating the bias in trained models (Schick et al., 2021). We recognize that this is not a straightforward task with a one-size-fits all solution, but we propose that as much attention should be dedicated to the corpora used for training language models as to the models themselves, and that corpora transparency is a prerequisite for language model accountability.

References

Abid, A., Farooqi, M., and Zou, J. (2021). Persistent Anti-Muslim Bias in Large Language Models. *arXiv preprint arXiv:2101.05783*.

Aluru, S. S., Mathew, B., Saha, P., and Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Badjatiya, P., Gupta, S., Gupta, M., and Varma, V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760.

Basile, V., Bosco, C., Fersini, E., Debora, N., Patti, V., Pardo, F. M. R., Rosso, P., Sanguinetti, M., et al. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Bender, E., Gebru, T., McMillan-Major, A., et al. (2021). On the dangers of stochastic parrots: Can language models be too big. *Proceedings of FAccT*.

Bender, E. M. and Friedman, B. (2018). Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Bhardwaj, R., Majumder, N., and Poria, S. (2021). Investigating gender bias in BERT. *Cognitive Computation*, pages 1–11.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Bordia, S. and Bowman, S. R. (2019). Identifying and reducing gender bias in word-level language models. *arXiv preprint arXiv:1904.03035*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. (2020). Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*.

Caswell, I., Kreutzer, J., Wang, L., Wahab, A., van Esch, D., Ulzii-Orshikh, N., Tapo, A., Subramani, N., Sokolov, A., Sikasote, C., et al. (2021). Quality at a glance: An audit of web-crawled multilingual datasets. *arXiv preprint arXiv:2103.12028*.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

de Lima, L. H. C., Reis, J., Melo, P., Murai, F., and Benevenuto, F. (2021). Characterizing (un) moderated textual data in social systems. *arXiv preprint arXiv:2101.00963*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Engel, J., Hoffman, M., and Roberts, A. (2017). Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*.

Farrell, T., Fernandez, M., Novotny, J., and Alani, H. (2019). Exploring misogyny across the manosphere in reddit. In *Proceedings of the 10th ACM Conference on Web Science*, pages 87–96.

Florez, O. U. (2019). On the unintended social bias of training language generation models with data from local media. *arXiv preprint arXiv:1911.00461*.

Fortuna, P. and Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Foubert, J. D., Blanchard, W., Houston, M., and Williams, R. R. (2019). Pornography and sexual violence. In *Handbook of Sexual Assault and Sexual Assault Prevention*, pages 109–127. Springer.

Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., and Kourtellis, N. (2018). Large scale crowdsourcing and characterization of Twitter abusive behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Fritz, N., Malic, V., Paul, B., and Zhou, Y. (2020). Worse than objects: The depiction of black women and men and their sexual relationship in pornography. *Gender Issues*, pages 1–21.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., and Crawford, K. (2018). Datasheets for datasets. *arXiv preprint arXiv:1803.09010*.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. (2020). Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Gonen, H. and Goldberg, Y. (2019). Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Hammami, M., Chahir, Y., and Chen, L. (2003). Webguard: Web based adult content detection and filtering system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 574–578. IEEE.

Hashemi, M. (2020). Web page classification: a survey of perspectives, gaps, and future directions. *Multimedia Tools and Applications*, pages 1–25.

Ho, W. H. and Watters, P. A. (2004). Statistical and structural approaches to filtering internet pornography. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 5, pages 4792–4798. IEEE.

Hube, C. (2017). Bias in Wikipedia. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 717–721.

Hutchinson, B., Prabhakaran, V., Denton, E., Webster, K., Zhong, Y., and Denuyl, S. (2020). Social biases in NLP models as barriers for persons with disabilities. *arXiv preprint arXiv:2005.00813*.

Jo, E. S. and Gebru, T. (2020). Lessons from archives: strategies for collecting sociocultural data in machine learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 306–316.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kiritchenko, S. and Mohammad, S. M. (2018). Examining gender and race bias in two hundred sentiment analysis systems. *arXiv preprint arXiv:1805.04508*.

Kolias, V., Anagnostopoulos, I., and Kayafas, E. (2014). Exploratory analysis of a terabyte scale web corpus. *arXiv preprint arXiv:1409.5443*.

Manzini, T., Lim, Y. C., Tsvetkov, Y., and Black, A. W. (2019). Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*.

Massanari, A. (2017). # gamergate and the fappening: How reddit's algorithm, governance, and culture support toxic technocultures. *New media & society*, 19(3):329–346.

Mathew, B., Illendula, A., Saha, P., Sarkar, S., Goyal, P., and Mukherjee, A. (2019). Temporal effects of unmoderated hate speech in gab. *arXiv preprint arXiv:1909.10966*.

Matic, S., Iordanou, C., Smaragdakis, G., and Laoutaris, N. (2020). Identifying sensitive URLs at web-scale. In *Proceedings of the ACM Internet Measurement Conference*, pages 619–633.

May, C., Wang, A., Bordia, S., Bowman, S. R., and Rudinger, R. (2019). On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*.

Mishra, P., Yannakoudakis, H., and Shutova, E. (2019). Tackling online abuse: A survey of automated abuse detection methods. *arXiv preprint arXiv:1908.06024*.

Nadeem, M., Bethke, A., and Reddy, S. (2020). Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Ortiz Suarez, P. J., Sagot, B., and Romary, L. (2019). Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut fur Deutsche Sprache.

Paullada, A., Raji, I. D., Bender, E. M., Denton, E., and Hanna, A. (2020). Data and its (dis) contents: A survey of dataset development and use in machine learning research. *arXiv preprint arXiv:2012.05345*.

Polpinij, J., Chotthanom, A., Sibunruang, C., Chamchong, R., and Puangpronpitag, S. (2006). Content-based text classifiers for pornographic web filtering. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1481–1485. IEEE.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Rowley, H. A., Jing, Y., and Baluja, S. (2006). Large scale image-based adult-content filtering. *Google Research Paper*.

Roziewski, S. and Stokowiec, W. (2016). Languagecrawl: A generic tool for building language models upon common-crawl. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2789–2793.

Schick, T., Udupa, S., and Schütze, H. (2021). Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *arXiv preprint arXiv:2103.00453*.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International workshop on natural language processing for social media*, pages 1–10.

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. (2019). The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*.

Shim, J. W., Kwon, M., and Cheng, H.-I. (2015). Analysis of representation of sexuality on women's and men's pornographic websites. *Social Behavior and Personality: an international journal*, 43(1):53–62.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J. W., Kreps, S., et al. (2019). Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Sweeney, C. and Najafian, M. (2019). A transparent framework for evaluating unintended demographic bias in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1662–1667.

Tadesse, M. M., Lin, H., Xu, B., and Yang, L. (2019). Detection of depression-related posts in reddit social media forum. *IEEE Access*, 7:44883–44893.

Vidgen, B. and Derczynski, L. (2020). Directions in abusive language training data: Garbage in, garbage out. *arXiv preprint arXiv:2004.01670*.

Wachs, S., Wright, M. F., and Vazsonyi, A. T. (2019). Understanding the overlap between cyberbullying and cyberhate perpetration: Moderating effects of toxic online disinhibition. *Criminal Behaviour and Mental Health*, 29(3):179–188.

Wang, Z. J., Choi, D., Xu, S., and Yang, D. (2021). Putting humans in the natural language processing loop: A survey. *arXiv preprint arXiv:2103.04044*.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Wehrmann, J., Simões, G. S., Barros, R. C., and Cavalcante, V. F. (2018). Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing*, 272:432–438.

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, É. (2020). CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012.

Wolf, M. J., Miller, K. W., and Grodzinsky, F. S. (2017). Why we should have seen that coming: comments on microsoft's tay "experiment," and wider implications. *The ORBIT Journal*, 1(2):1–12.

World Bank (2018). Indiviuals using the Internet. https://data.worldbank.org/indicator/IT.NET.USER.ZS?end=2017&locations=US&start=2015. Accessed: 2021-01-10.

Zhang, Y., Wang, G., Li, C., Gan, Z., Brockett, C., and Dolan, B. (2020). Pointer: Constrained text generation via insertion-based generative pre-training. *arXiv preprint arXiv:2005.00558*.

Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., and Chang, K.-W. (2019). Gender bias in contextualized word embeddings. *arXiv preprint arXiv:1904.03310*.

Zhao, J., Zhou, Y., Li, Z., Wang, W., and Chang, K.-W. (2018). Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496*.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Continual Quality Estimation with Online Bayesian Meta-Learning

Abiola Obamuyide[1] **Marina Fomicheva**[1] **Lucia Specia**[1,2]
[1]Department of Computer Science, University of Sheffield
[2]Department of Computing, Imperial College London
United Kingdom
{a.obamuyide,m.fomicheva,l.specia}@sheffield.ac.uk

Abstract

Most current quality estimation (QE) models for machine translation are trained and evaluated in a static setting where training and test data are assumed to be from a fixed distribution. However, in real-life settings, the test data that a deployed QE model would be exposed to may differ from its training data. In particular, training samples are often labelled by one or a small set of annotators, whose perceptions of translation quality and needs may differ substantially from those of end-users, who will employ predictions in practice. To address this challenge, we propose an online Bayesian meta-learning framework for the continuous training of QE models that is able to adapt them to the needs of different users, while being robust to distributional shifts in training and test data. Experiments on data with varying number of users and language characteristics validate the effectiveness of the proposed approach.

1 Introduction

Quality Estimation (QE) models aim to evaluate the output of Machine Translation (MT) systems at run-time, when no reference translations are available (Blatz et al., 2004; Specia et al., 2009). QE models can be applied for instance to improve translation productivity by selecting high-quality translations amongst several candidates. A number of approaches have been proposed for this task (Specia et al., 2009, 2015; Kim et al., 2017; Kepler et al., 2019; Ranasinghe et al., 2020), and a shared task yearly benchmarks proposed approaches (Fonseca et al., 2019; Specia et al., 2020).

Different users of MT output have varying quality needs and standards, depending for instance on the downstream task at hand, or the level of their knowledge of the languages involved, and training for the task. Thus, the perception of the quality of MT output can be subjective, and therefore the quality estimates obtained from a model trained on data from one set of users may not serve the needs of a different set users. However, most existing QE models are trained and evaluated in a static setting which assumes a fixed distribution of train and test data. This consequently leads to suboptimal performance when faced with test data from a different set of users in practice.

The few previous approaches to develop QE models that are able to learn from a continuous stream of data suffer from the following limitations: they do not have an explicit objective that encourages the model to exploit common structures shared among different users to continually adapt efficiently for new users (Turchi et al., 2014), or assume a fixed number of users, and that the identity of each user is known in advance (de Souza et al., 2015). In addition, these previous approaches do not explicitly account for the underlying uncertainties in the data in order to improve performance.

In contrast, we propose a continual meta-learning framework that makes none of the aforementioned assumptions, but instead considers each user as a task and explicitly meta-learns the common structure shared among different users. This approach further exploits the underlying uncertainties in the streaming data through Bayesian inference to improve performance. In addition, the proposed approach is applicable even in a setting where no user identities are available, for instance due to privacy concerns, but where we still want to learn and adapt as efficiently as possible from supervision data that arrives incrementally.

2 Background

2.1 Continual Learning

Continual learning (Ring, 1994; Thrun, 1996; Zhao and Schmidhuber, 1996) aims to develop mod-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 190–197
August 1–6, 2021. ©2021 Association for Computational Linguistics

els that are capable of learning from a continuous stream of sequential tasks, $\mathcal{T}_1, \mathcal{T}_2, .., \mathcal{T}_T$, with each task $\mathcal{T}_t$ having its associated train $\mathcal{D}_t^{train}$, validation $\mathcal{D}_t^{val}$ and test $\mathcal{D}_t^{test}$ splits. A major challenge associated with learning in this setting is the issue of *catastrophic forgetting*, where a model forgets knowledge of how to perform previous tasks as new tasks are encountered. Most recent work in lifelong learning has focused on ways of mitigating catastrophic forgetting, and approaches proposed include replay-based methods (Rebuffi et al., 2017; Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019), which replay either stored or generated samples to remind the model of how to perform previous tasks; regularization-based methods (Kirkpatrick et al., 2017; Zenke et al., 2017), which utilize an additional regularization term to enforce retaining knowledge learned from previous tasks; and parameter-isolation methods, which make use of dedicated parameters for each task to prevent interference among tasks (Rusu et al., 2016; Fernando et al., 2017). Lange et al. (2019) presents an overview of recent continual learning methods. Research in continual learning can generally be carried in one of two settings (Aljundi et al., 2019): in a *task-incremental* continual learning setting, where the learner is sequentially given access to all the data of each task and is allowed to make multiple passes over it, with task boundaries and identities known to the learner; or in an *online continual learning* setting, where the learner is only allowed a single pass over the data of each task, and with no task identities or boundaries known to the learner. In this work we conduct experiments in the online continual learning setting.

2.2 Meta-Learning

The goal of meta-learning, also known as learning to learn (Schmidhuber, 1987; Thrun and Pratt, 1998), is to develop models that can learn more efficiently over time, by generalizing from knowledge of how to solve related tasks from a given distribution of tasks. Given a learner model $f_{\boldsymbol{w}}$, for instance a neural network parametrized by $\boldsymbol{w}$, and a distribution $p(\mathcal{T})$ over tasks $\mathcal{T}$, gradient-based meta-learning approaches such as MAML (Finn et al., 2017) seek to learn the parameters of the learner model which can be quickly adapted to new tasks sampled from the same distribution of tasks. In formal terms, these approaches seek parameters that optimize the meta-objective:

$$\min_{\boldsymbol{w}} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})} \left[\mathcal{L}_{\mathcal{T}} \left(\mathcal{U}_k \left(\boldsymbol{w}; \mathcal{D}_{\mathcal{T}}\right)\right)\right] \quad (1)$$

where $\mathcal{L}_{\mathcal{T}}$ is the loss and $\mathcal{D}_{\mathcal{T}}$ is training data from task $\mathcal{T}$, and $\mathcal{U}_k$ denotes k steps of a gradient descent learning rule such as SGD.

In order to account for uncertainty and improve robustness, Bayesian approaches to meta-learning have also been proposed (Kim et al., 2018; Finn et al., 2018; Ravi and Beatson, 2019; Wang et al., 2020; Nguyen et al., 2020).

2.3 Meta-Learning for Continual Learning

Meta-learning for continual learning methods generally make use of the meta-learning objective one task at a time to ensure that learning on the current task does not lead to catastrophic forgetting on previous tasks. For instance, both Riemer et al. (2019) and Obamuyide and Vlachos (2019) propose to combine REPTILE (Nichol and Schulman, 2018), a first order meta-learning algorithm, together with experience replay to improve performance during continual learning. Javed and White (2019) proposed an online-aware meta-learning (OML) objective for learning representations that are less prone to catastrophic forgetting during continual learning. Holla et al. (2020) proposed to combine the OML objective together with experience replay for improved continual learning performance. Recently, Gupta et al. (2020) proposed Look-Ahead MAML (LA-MAML), which meta-learns per-parameter learning rates to help adapt to changing data distributions during continual learning.

These approaches have demonstrated that meta-learning can yield performance improvements for continual learning. Our work builds on these approaches and additionally demonstrates that the performance of meta-learning for continual learning can be further improved with the combination of an adaptive learning rate and Bayesian inference.

2.4 Bayesian Inference with Stein Variational Gradient Descent

Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016) is a Bayesian inference method which works by initializing a set of samples, also known as particles, from a simple distribution and iteratively updating the particles to match samples from a target distribution. Because its particle update rule is deterministic and differentiable, it can

be used to perform Bayesian inference in the meta-learning inner loop, since the entire update process can still be differentiated through for gradient-based updates from the outer loop.

In order to obtain N samples from a posterior $P(\boldsymbol{w})$, SVGD maintains N samples of model parameters, and iteratively transports the samples to match samples from the target distribution. Let the samples be represented by $\boldsymbol{W} = \{\boldsymbol{w}^n\}_{n=1}^N$. At each successive iteration t, SVGD updates each sample with the following update rule:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha_t \phi(\boldsymbol{w}_t) \tag{2}$$

where $\phi(\boldsymbol{w}_t) =$

$$\frac{1}{N}\sum_{n=1}^{N}\left[k(\boldsymbol{w}_t^n, \boldsymbol{w}_t)\nabla_{\boldsymbol{w}_t^n}\log p(\boldsymbol{w}_t^n) + \nabla_{\boldsymbol{w}_t^n} k(\boldsymbol{w}_t^n, \boldsymbol{w}_t)\right] \tag{3}$$

α_t is a step-size parameter and $k(.,.)$ is a positive-definite kernel, such as the RBF kernel.

Intuitively, the first term in Equation 3 implies that a particle determines its update direction through a weighted aggregate of the gradients from the other particles, with the kernel distance between the particles serving as the weight. Thus, closer particles have more weight in the aggregate. The second term of the equation can be understood as a repulsive force that prevents the particles from collapsing to a single point. For the case when the number of particles is one, the SVGD update procedure reduces to standard gradient ascent on the objective $p(\boldsymbol{w})$ for any kernel with the property $\nabla_{\boldsymbol{w}} k(\boldsymbol{w}, \boldsymbol{w}) = 0$, such as the RBF kernel. SVGD has been applied in a wide range of settings, including reinforcement learning (Liu et al., 2017; Haarnoja et al., 2017), uncertainty quantification (Zhu and Zabaras, 2018) and to improve performance in an offline meta-learning setup (Kim et al., 2018) which requires all tasks ahead of training. In this work we adapt SVGD to an online continual meta-learning setting for a natural language task.

3 Meta-Learning for Continual Learning with Adaptive SVGD

Learning continually from a stream of observations with varying underlying distributions involves dealing with various sources of uncertainty, which a model should properly account for in order to enhance its continual learning performance. One source of uncertainty is in the learning rate, that is, how fast learning should proceed on new data in order to both reduce catastrophic forgetting and enhance performance on the current task. Another source is the inherent uncertainty in the values of the model's parameters themselves. Learning an adaptive learning rate, for instance as proposed in Gupta et al. (2020), can help account for the first source of uncertainty, and Bayesian inference can be used to help a model account for the other source of uncertainty. In order to properly model both sources of uncertainty during continual learning, we propose to both perform inference of model parameters with SVGD, and meta-learn an adaptive per-parameter learning rate for SVGD updates. Thus, the SVGD update in Equation 2 now becomes:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \boldsymbol{\alpha}_t \cdot \phi(\boldsymbol{w}_t) \tag{4}$$

where $\boldsymbol{\alpha}_t$ is a learnable parameter containing per-parameter learning rates, and $\cdot$ is the dot product.

The aim is then to meta-learn both the parameters of the model and the per-parameter learning rates that enhance continual learning performance. The advantage of this approach is that it allows for greater flexibility to adapt to non-stationary data distributions during continual learning. In the experiments, we demonstrate that this change leads to improved performance for the task of continual quality estimation. The proposed approach is illustrated in Algorithm 1.

We first initialize the parameters of the QE model, and the learning rate (line 1). Then for each mini-batch in a task t that arrives, we store its training instances in the buffer with a probability p (lines 2-6). In the inner loop, we perform K SVGD updates (using Equation 4) starting from the initial model parameters W_0 (lines 7-9). In the outer loop, instances in the current mini-batch are augmented with instances sampled from the buffer (line 10). Finally, the augmented mini-batch is used to perform a meta-update on the learning rate (line 11), and on the parameters of the QE model (line 12). Because this approach can also be considered the online counterpart to the Bayesian Model Agnostic Meta-Learning approach of Kim et al. (2018), we refer to it as Continual Quality Estimation with Online Bayesian Meta-Learning (*CQE-OBML*).

4 Experiments and Results

The QT21 Dataset We evaluate our approach with the publicly available **QT21** (Specia et al., 2017), a large-scale dataset containing translations

Algorithm 1 Continual Quality Estimation with Online Bayesian Meta-Learning (*CQE-OBML*)

Require: QE model f_{W_0}, learning rates $\boldsymbol{\alpha}_0$, β
Require: Buffer B, update probability p

Initialize W_0 , $\boldsymbol{\alpha}_0$
for t = 1,2,3,... **do**
  for each (X_t, Y_t) in D_t^{train} **do**
    if $random() < p$ **then**
      Update $B \leftarrow B \cup (X_t, Y_t)$
    end if
    for k = 1,..K **do**
      $W_k = SVGD(W_{k-1}, \boldsymbol{\alpha}_0, X_t, Y_t)$
    end for
    $(X_v, Y_v) \leftarrow (X_t, Y_t) \cup \textit{sample}(B)$
    $\boldsymbol{\alpha}_0 \leftarrow \boldsymbol{\alpha}_0 - \beta \nabla_{\boldsymbol{\alpha}_0} \mathcal{L}_t \left(f_{W_k}(X_v), Y_v\right)$
    $W_0 \leftarrow W_0 - \boldsymbol{\alpha}_o \cdot \nabla_{W_0} \mathcal{L}_t \left(f_{W_k}(X_v), Y_v\right)$
  end for
end for

PE ID	Train	Dev	Test
PE1	1440	360	200
PE2	2160	540	300
PE3	1444	361	195
PE4	1834	459	244
PE5	4866	1217	617
PE6	1677	420	203
PE7	1567	392	241
Total	14988	3749	2000

(a) QT21 en-lv (nmt)

PE ID	Train	Dev	Test
PE1	9952	2488	559
PE2	3445	862	193
PE3	8770	2193	537
PE4	4579	1145	276
PE5	7651	1913	435
Total	34397	8601	2000

(b) QT21 en-cs (smt)

Table 1: Number of instances per post-editor (PE) for the QT21 dataset.

from both statistical (smt) and neural (nmt) machine translation systems in multiple language directions.[1] This is the largest dataset with annotator information available. We use data from the English-Latvian (en-lv) and English-Czech (en-cs) language pairs. These languages were chosen as they contain the largest number of annotators. Each instance in the dataset is a tuple of source sentence, its machine translation, the corresponding post-edited translation by a professional translator (post-editor), a reference translation and other information such as (anonymized) post-editor identifier. We construct a QE dataset from this corpus by computing the HTER (Snover et al., 2006) values between each source sentence and its post-edited translation. We thereafter split the data into train, dev and test splits for each post-editor. A breakdown of the number of train, dev and test instances per post-editor is available in Table 1.

Benchmark Approaches SEQUENTIAL is a baseline trained sequentially over the streaming data of each task. In each round, the model parameters are initialized from that of the previous round; A-GEM (Chaudhry et al., 2019) is a continual learning method which utilizes the gradients of samples of previous tasks saved in a buffer as an optimization constraint to prevent catastrophic forgetting; OML-ER (Holla et al., 2020) augments the Online-Aware Meta-Learning approach of Javed and White (2019) with experience replay from a buffer; LA-MAML (Gupta et al., 2020) learns per-parameter learning rates using meta-learning; MTL-IID is trained on the concatenated and shuffled data from all users for multiple epochs in multi-task fashion. It assumes i.i.d access to the data from all users, and thus serves as an upper-bound for the performance.

QE Model The quality estimation model used by all continual learning methods is based on multilingual DistilBERT (Sanh et al., 2019), a smaller version of multi-lingual BERT (Devlin et al., 2019) trained with knowledge distillation (Buciluǎ et al., 2006; Hinton et al., 2015). It accepts as input the source and machine translation outputs concatenated as a single text, separated by a '[SEP]' token and prepended with a '[CLS]' token. The representation of the '[CLS]' token is then passed to a linear layer to predict HTER (Snover et al., 2006) values as regression targets.

Evaluation We report Pearson's r correlation scores and Mean Absolute Error (MAE) between model output and gold labels, both standard evaluation metrics in QE.

Each experiment is repeated across five (5) different orders of the tasks and five (5) different random seeds, and we report the average.

4.1 Comparison with Benchmark Approaches

The results of our approach in comparison with other benchmark approaches are presented in Table 2. We can observe that naively training sequentially on the data of each task as it arrives (SEQUENTIAL) leads to poor results.

[1] http://www.qt21.eu/resources/data/

Method	en-lv		en-cs	
	Pearson ↑	MAE ↓	Pearson ↑	MAE ↓
MTL-IID	59.17	0.1450	54.79	0.1547
SEQUENTIAL	47.07	0.1773	50.08	0.1689
A-GEM	46.29	0.1794	46.49	0.1736
OML-ER	52.58	0.1621	50.40	0.1635
LA-MAML	52.86	0.1621	50.56	0.1631
CQE-OBML	**53.67**	**0.1596**	**51.19**	**0.1619**

Table 2: Comparison with benchmark approaches.

OML-ER outperforms both SEQUENTIAL and A-GEM, likely because of its combination of meta-learning and experience replay, which makes it better able to combat forgetting. LA-MAML slightly improves over the results of OML-ER, as a result of its meta-learned learning rate. We find that our approach, CQE-OBML, which combines a meta-learned adaptive learning rate together with Bayesian inference, outperforms previous approaches. This demonstrates the effectiveness of adequately modelling the various sources of uncertainty in continual meta-learning.

4.2 Analysis of Model Components

We investigate the effect of the various components of our approach through an ablation study. As shown in Table 3, our approach (CQE-OBML) without the adaptive learning rate (-LR (α)) has a drop in performance, especially for en-cs. Without inference with SVGD (-SVGD), we observe a larger reduction in performance on both datasets, demonstrating the usefulness of incorporating Bayesian inference into the continual meta-learning of quality estimation models.

Method	en-lv		en-cs	
	Pearson ↑	MAE ↓	Pearson ↑	MAE ↓
CQE-OBML	53.67	0.1596	51.19	0.1619
- LR (α)	53.48	0.1598	50.94	0.1623
- SVGD	52.86	0.1621	50.56	0.1631

Table 3: Ablation of model components.

5 Conclusions

We proposed a framework for the continual meta-learning of machine translation quality estimation models, which is able to learn continually from the streaming data of multiple quality estimation users. We further incorporate an adaptive learning rate together with online Bayesian inference for improved performance. In experiments on quality estimation data from two language directions, we demonstrate improved performance over recent state-of-the-art continual learning methods.

Acknowledgements

This work was supported by funding from the Bergamot project (EU H2020 grant no. 825303).

References

Rahaf Aljundi, Klaas Kelchtermans, and T. Tuytelaars. 2019. Task-free continual learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11246–11255.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Chelsea Finn, Kelvin Xu, and S. Levine. 2018. Probabilistic model-agnostic meta-learning. In *Advances In Neural Information Processing Systems*.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

G. Gupta, Karmesh Yadav, and Liam Paull. 2020. Look-ahead meta learning for continual learning. In *Advances In Neural Information Processing Systems (NeurIPS)*.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Nithin Holla, Pushkar Mishra, H. Yannakoudakis, and Ekaterina Shutova. 2020. Meta-learning with sparse experience replay for lifelong language learning. *ArXiv*, abs/2009.04891.

Khurram Javed and Martha White. 2019. Meta-learning representations for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. de Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1820–1830. Curran Associates, Inc.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 562–568. Association for Computational Linguistics.

Taesup Kim, Jaesik Yoon, O. Dia, S. Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. In *Advances In Neural Information Processing Systems*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and Others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Maarit Koponen, W. Aziz, L. Ramos, Lucia Specia, J. Rautio, M. González, L. Carlson, and C. España-Bonet. 2012. Post-editing time as a measure of cognitive effort. In *AMTA 2012 Workshop on Post-editing Technology and Practice (WPTP)*.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. *CoRR*, abs/1909.08383.

Qiang Liu and Dilin Wang. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems 29*, pages 2378–2386. Curran Associates, Inc.

Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. 2017. Stein variational policy gradient. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*.

Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. 2020. Uncertainty in model-agnostic meta-learning using variational inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3090–3100.

Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(2):1.

Abiola Obamuyide and Andreas Vlachos. 2019. Meta-learning improves lifelong relation extraction. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 224–229, Florence, Italy. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Sachin Ravi and Alex Beatson. 2019. Amortized bayesian meta-learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 2001–2010.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2019. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Mark Bishop Ring. 1994. *Continual learning in reinforcement environments*. Ph.D. thesis, University of Texas at Austin Austin, Texas 78712.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, MA.

José Guilherme Camargo de Souza, Matteo Negri, Elisa Ricci, and Marco Turchi. 2015. Online multitask learning for machine translation quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 219–228. The Association for Computer Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Kim Harris, Aljoscha Burchardt, Marco Turchi, Matteo Negri, and Inguna Skadina. 2017. Translation quality and productivity: A study on rich morphology languages. In *Machine Translation Summit XVI*, pages 55–71.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, System Demonstrations*, pages 115–120. The Association for Computer Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Sebastian Thrun. 1996. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, pages 640–646.

Sebastian Thrun and Lorien Pratt. 1998. Learning to Learn: Introduction and Overview. In *Learning to Learn*, pages 3–17. Springer US, Boston, MA.

Marco Turchi, Antonios Anastasopoulos, José G. C. de Souza, and Matteo Negri. 2014. Adaptive quality estimation for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 710–720. The Association for Computer Linguistics.

Zhenyi Wang, Yang Zhao, Ping Yu, Ruiyi Zhang, and Changyou Chen. 2020. Bayesian meta sampling for fast uncertainty adaptation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual Learning Through Synaptic Intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995, International Convention Centre, Sydney, Australia.

Jieyu Zhao and Jurgen Schmidhuber. 1996. Incremental self-improvement for life-time multi-agent reinforcement learning. In *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, Cambridge, MA*, pages 516–525.

Yinhao Zhu and Nicholas Zabaras. 2018. Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.*, 366:415–447.

A Additional Results

We present additional results on the **WPTP12** dataset (Koponen et al., 2012),which is a small English-Spanish (en-es) translation dataset consisting of documents from the news domain. It features translations from eight different machine translation systems. Each instance in the dataset includes the corresponding post-edited translation along with post-editing time and HTER scores computed between the translation and the corresponding post-edit. Statistics about the number of instances per post-editor are in Table 4.

Table 5 contains the results obtained on this dataset. As a result of its size, all methods generally find it challenging, with reduced performance across-the-board. Despite reduced performance in terms of mean absolute error, our approach obtains better Pearson correlation than all previous methods.

PE ID	Train	Dev	Test
A1	121	40	42
A2	121	40	42
A3	121	40	42
A4	121	40	42
A5	121	40	42
A6	121	40	42
A7	121	40	42
A8	121	40	42
Total	968	320	336

Table 4: Number of instances per Post Editor (PE) for the WPTP12 dataset.

Method	WPTP12	
	Pearson ↑	MAE ↓
SEQUENTIAL	33.05	0.2061
A-GEM	38.95	0.2066
OML-ER	39.17	0.1786
LA-MAML	38.89	**0.1772**
CQUEST-OBML	**40.11**	0.1780

Table 5: Averaged performance for all methods.

B Additional Experimental Details

All models make use of the same values for hyper-parameters such as learning rate and batch size, selected by manual search in initial experiments. These are provided in Table 6.

Hyper-parameter	**Value**
Learning rate	3e-5
Mini-batch size	16
Max. sequence length	100

Table 6: Hyper-parameter values for all compared approaches

A Span-based Dynamic Local Attention Model for Sequential Sentence Classification

Xichen Shang, Qianli Ma*, Zhenxi Lin, Jiangyue Yan, Zipeng Chen
School of Computer Science and Engineering, South China University of Technology
Key Laboratory of Big Data and Intelligent Robot
(South China University of Technology), Ministry of Education
shangxichen@foxmail.com, qianlima@scut.edu.cn*

Abstract

Sequential sentence classification aims to classify each sentence in the document based on the context in which sentences appear. Most existing work addresses this problem using a hierarchical sequence labeling network. However, they ignore considering the latent segment structure of the document, in which contiguous sentences often have coherent semantics. In this paper, we proposed a span-based dynamic local attention model that could explicitly capture the structural information by the proposed supervised dynamic local attention. We further introduce an auxiliary task called span-based classification to explore the span-level representations. Extensive experiments show that our model achieves better or competitive performance against state-of-the-art baselines on two benchmark datasets.

1 Introduction

The goal of Sequential Sentence Classification (SSC) is to classify each sentence in a document based on rhetorical structure profiling process (Jin and Szolovits, 2018), and the rhetorical label of each sentence is related to the surrounding sentences, which is different from the general sentence classification that does not involve context. An example is shown in Figure 1, the document is divided into rhetorical labels such as "background" and "outcome" for five sentences in NICTA dataset. The SSC task is crucial for downstream domains such as information retrieval (Edinger et al., 2017), question answering (Cohen et al., 2018) and so on.

Traditional statistical methods, such as HMM (Lin et al., 2006), CRF (Hirohata et al., 2008; Hassanzadeh et al., 2014), etc., heavily rely on numerous carefully hand-designed features. In contrast, recent methods based on end-to-end neural networks utilize hierarchical sequence encoders followed by the CRF layer to contextualize sentence representations, which achieved promising results. The first neural network approach (Lee and Dernoncourt, 2016) combined RNN with CNN that incorporates preceding sentences to encode the contextual content and further use a CRF layer to optimize the predicted label sequence. Recently, Jin and Szolovits (2018) propose a hierarchical sequential labeling network to make use of the contextual information within surrounding sentences to help classify. Conversely, Cohan et al. (2019) employ BERT (Devlin et al., 2018) to capture contextual dependencies without hierarchical encoding or CRF layer. Yamada et al. (2020) introduce Semi-Markov CRFs (Ye and Ling, 2018) to assign a rhetorical label at span-level rather than single sentence.

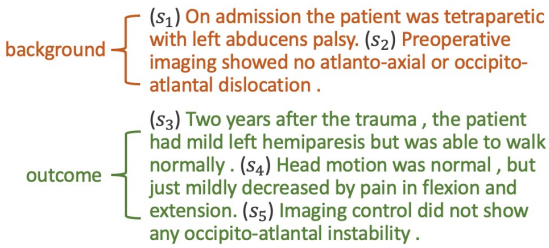


Figure 1: An example of NICTA dataset for SSC task. The text has five sentences and is divided into two segments $\{(s_1, s_2), (s_3, s_4, s_5)\}$ by labels.

Nevertheless, the above-mentioned methods ignore the latent structural information (e.g. segmentation) in the document, which is the grouping of content into topically coherent segments. Intuitively, a segment with several continuous sentences is expected to be more coherent semantics than the text spanning different segments, e.g., the text with two segments in Figure 1. In this paper, we propose a novel span-based dynamic local attention model to explore the latent segment structure in a document for SSC task. First, we introduce

*Corresponding author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 198–203
August 1–6, 2021. ©2021 Association for Computational Linguistics

dynamic local attention guided by segmentation supervision signal to focus on the surrounding sentences with coherent semantics, called Supervised Dynamic Local Attention (SDLA). Furthermore, we introduce an auxiliary task called span-based classification, which calculates semantic representations of spans and performs span classification on them to obtain predicted rhetorical labels. The dynamic local attention mechanism and the auxiliary task complement each other to enhance the model capacity to perceive segment structure and improve the performance of SSC task. The results on two benchmark datasets show that our method achieves better or competitive performance than state-of-the-art baselines.

2 Proposed Method

In this paper, we propose a Span-based Dynamic Local Attention Model for sequential sentence classification with two novel components: supervised dynamic local attention and auxiliary span-based classification task, respectively. The architecture of our model is shown in Figure 2.

2.1 Sentence Representations

For SSC task, given a sequence of sentences $X = \{x_1, x_2, \cdots, x_N\}$, the model needs to predict the label of each sentence $Y = \{y_1, y_2, \cdots, y_N\}$ based on the context which the sentence appears, where N is the number of sentences. Following the previous work (Yamada et al., 2020), we first feed each sentence into BERT pre-trained with PubMed (Peng et al., 2019) and then extract the encoding corresponding to [CLS] token as sentence encoding $S = \{s_1, s_2, \cdots, s_N\}$ (we implement it using Sentence-BERT (Reimers and Gurevych, 2019)). Then, we employ two bidirectional LSTM layers to produce context-informed sentence representation $h_i^c \in \mathbf{R}^d$ for whole document :

$$H^c = \{h_1^c, h_2^c, \cdots, h_N^c\} \tag{1}$$

2.2 Supervised Dynamic Local Attention

In this section, we introduce dynamic local attention guided by a supervised segmentation signal to learn latent segment structure in a document. Firstly, we generate the sentence-level attention spans for each sentence by training soft masking (Nguyen et al., 2020), using pointing mechanism (Vinyals et al., 2015) to approximate left and right boundary positions of the mask vector. Given the query Q and key K, where $Q = K = H^c$,

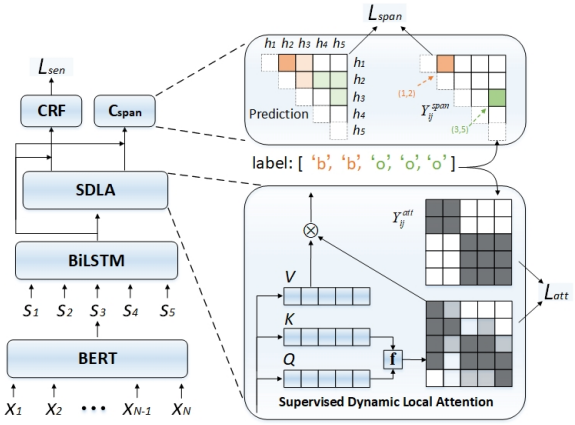


Figure 2: The overview of our model, exemplified by the sample in Figure 1. The labels 'b' and 'o' stand for "background" and "outcome", respectively. C_{span} denotes Auxiliary Span-based Classification Task.

we calculate the left and right boundary matrix $\hat{\phi}_l, \hat{\phi}_r \in R^{N \times N}$ for query Q as follows:

$$\hat{\phi}_l = \mathcal{S}(\frac{Q^T W_L^Q (K W_L^K)^T}{\sqrt{d}} \odot M) \tag{2}$$

$$\hat{\phi}_r = \mathcal{S}(\frac{Q^T W_R^Q (K W_R^K)^T}{\sqrt{d}} \odot M^T) \tag{3}$$

$$M_{ij} = \begin{cases} -\infty, & i < j \\ 1, & i \geq j \end{cases} \tag{4}$$

where $\mathcal{S}$ is the softmax function, $\odot$ is element-wise product, and $W_L^Q, W_L^K, W_R^Q, W_R^Q \in \mathbf{R}^{d \times d}$ are trainable parameters. Eq. (2)-(3) approximate the left and right boundary positions of the mask matrix for the query Q (Each row approximate the mask vector of the entire document corresponding to each sentence in sequence order). Note that we additionally introduce mask matrix M to ensure that the left boundary position l and the right boundary position r generated at position i satisfy this relationship such that $0 \leq l \leq i \leq r \leq N$.

Given the above definitions, the attention span masking matrix M_a can be achieved by compositing the left and right boundary matrix :

$$M_a = (\hat{\phi}_l L_N) \odot (\hat{\phi}_r L_N^T) \tag{5}$$

where $L_N \in \{0, 1\}^{N \times N}$ denotes a unit-value (1) upper-triangular matrix.

Then we combine self-attention with the attention span masking, enabling the model to focus on semantically related sentences around the target

position and eliminate noisy aggregations :

$$A = \frac{(QW^Q)(KW^K)^T}{\sqrt{d}} \odot M_a \quad (6)$$

$$H^{att} = \mathcal{S}(A)(H^c W^H) \quad (7)$$

where W^Q, W^K, W^H are the trainable parameters.

However, in the absence of a supervised process, the dynamic local attention may fail to focus on the corresponding informative sentences of the target, especially for limited data, so we further introduce the segmentation signal to guide the learning of dynamic local attention to capture coherent semantics more accurately. Specifically, we employ binary cross-entropy loss to describe the differences between attention matrix A and segment signal Y^{att}:

$$\mathcal{L}_{att} = BCE(\sigma(A), Y^{att}) \quad (8)$$

$$Y^{att}_{ij} = \begin{cases} 1, & E_{ij} = 1 \\ 0, & else \end{cases} \quad (9)$$

where σ is sigmoid function. $E_{ij} = 1$ denotes i-th sentence and j-th are in the same segment (e.g. (s_1, s_2) and (s_4, s_5) in Figure 1).

Finally, we concatenate H^c and H^{att} as the contextual representations H and add a CRF layer to classify each sentence.

2.3 Auxiliary Span-based Classification Task

Due to the obvious label consistency of sentences within spans, we introduce an additional auxiliary task called span-based classification to improve the performance at the span-level. To this effect, we consider all possible spans of various lengths and propose a tagging scheme for span-based classification. The scheme uses the same labels as sentence-level to represent the label of a span. Firstly, we represent a span from the i-th sentence to the j-th sentence as a vector h_{ij}, which is concatenated by four-vectors similar to Zhao et al. (2020):

$$h_{ij} = \{h_i; h_j; \hat{h}_{i:j}; \varphi(j - i + 1)\} \quad (10)$$

where $\hat{h}_{i:j}$ is the attention output over the final sentence representation H in the span, and $\varphi(j - i + 1)$ is the feature vector encoding the span size.

We employ a cross-entropy category loss for span-based classification:

$$\mathcal{L}_{span} = CE(\hat{Y}^{span}, Y^{span}) \quad (11)$$

$$Y^{span}_{ij} = \begin{cases} label, & F_{ij} = 1 \\ 0, & else \end{cases} \quad (12)$$

where $\hat{Y}^{span}$ is the output probability at span-level, F_{ij} denotes i-th sentence and j-th sentence (i, j satisfy the relationship $i < j$) are in the same segment and i, j is the beginning and end of the segment respectively (e.g. (s_1, s_2) and (s_3, s_5) in Figure 1).

2.4 Objective Function

The overall objective function includes cross-entropy loss $\mathcal{L}_{sen}$, $\mathcal{L}_{span}$ for sentence and span-based classification and supervised attention loss $\mathcal{L}_{att}$:

$$\mathcal{L} = \mathcal{L}_{sen} + \lambda_{att}\mathcal{L}_{att} + \lambda_{span}\mathcal{L}_{span} \quad (13)$$

where $\lambda_{att}, \lambda_{span}$ are the hyperparameters for balancing the strength of $\mathcal{L}_{att}$ and $\mathcal{L}_{span}$.

3 Experiments

3.1 Experimental Setup

Datasets and Baselines To evaluate the effectiveness of our model, we conduct extensive experiments on two standard benchmark datasets from medical scientific abstracts, i.e. NICTA-PIBOSO (Kim et al., 2011) and PubMed 20k RCT (Dernoncourt and Lee, 2017). The detailed description of both datasets can be found in the appendix. We compare our model with three recent strong neural models, i.e., those of Jin and Szolovits (2018), Cohan et al. (2019), Yamada et al. (2020).

Implementation Details We set the size of hidden state to 200 and apply dropout with the probability of 0.5 for BiLSTM. Both hyperparameters λ_{att} and λ_{span} are set to 0.3. The batch size is 30. We use Adam optimizer with learning rate 0.003 and weight decay 0.99 for training. For evaluation, we maximize the score from sentence-level CRF to get the predicted labels of the corresponding se-

Models	Sentence-F1	Span-F1	P_k
NICTA-PIBOSO			
Jin and Szolovits (2018)	82.3	51.1	17.3
Cohan et al. (2019)	83.0	54.3	21.3
Yamada et al. (2020)	84.4	58.7	–
Ours	**86.8**	**62.9**	**12.2**
PubMed 20k RCT			
Jin and Szolovits (2018)	92.8	82.9	5.3
Cohan et al. (2019)	92.9	82.2	5.1
Yamada et al. (2020)	**93.1**	84.3	–
Ours	92.8	**84.5**	**4.1**

Table 1: The results comparison of our model and baselines on two benchmark datasets.

	background	other	intervention	study design	population	outcome
Avg Num. Sent.	2.8	2.6	1.3	1.0	1.1	5.2
Jin and Szolovits (2018)	53.5	34.0	31.7	64.1	70.8	51.4
Cohan et al. (2019)	55.5	41.0	36.9	63.0	69.9	57.4
Yamada et al. (2020)	60.5	**44.8**	34.3	62.4	72.9	64.3
Ours	**60.8**	35.4	**49.0**	**71.4**	**77.6**	**64.4**

Table 2: Average number of sentences in spans and Span-F1 scores for each rhetorical label on NICAT-PIBOSO.

	background	objective	methods	results	conclusions
Avg Num. Sent.	2.6	1.5	4.1	4.2	1.8
Jin and Szolovits (2018)	73.8	73.8	86.7	83.1	90.8
Cohan et al. (2019)	70.6	70.8	86.3	83.9	92.0
Yamada et al. (2020)	**74.7**	73.8	88.5	**85.8**	91.9
Ours	67.1	**74.4**	**89.3**	85.7	**93.2**

Table 3: Average number of sentences in spans and Span-F1 scores for each rhetorical label on PubMed 20k RCT.

quence. Following Yamada et al. (2020), we use Sentence-F1 and Span-F1 as evaluation metrics[1].

3.2 Experimental Results

Tabel 1 report the performance of our approaches against other methods on PubMed 20k RCT and NICTA-PIBOSO, respectively. The results of other methods are obtained from Yamada et al. (2020).

We can observe that our model, whether Sentence-F1 or Span-F1, is significantly better than other methods on NICTA-PIBOS, and we get a result comparable to Yamada et al. (2020) on PubMed 20k RCT. We believe that our model has remarkable performance on NICTA-PIBOS, which has fewer training samples but larger label space, because our model can capture latent segment structure by SDLA component and improve span representations by auxiliary span-based classification.

In addition, table 2 and 3 show the detail results of Span-F1 scores for each rhetorical label. Our model achieves better or similar performance than other baselines, except for "other" on NICAT-PIBOSO and "background" on PubMed 20k RCT. We speculate that the reason is that the sentence semantics corresponding to the "other" label are diverse and not significantly distinguishable from other labels, while the "background" usually appears before the "objective", and the sentence presentations of the two are easily confused.

3.3 Segmentation Performance Evaluation

Specially, if we ignore the rhetorical labels of sentences and only consider the segment boundaries (i.e. binary classification, whether it's a boundary), this can be regarded as text segmentation (Koshorek et al., 2018). We evaluate the segmentation performance of our model using the probabilistic P_k (Beeferman et al., 1999) error score (lower number, the better). The results[2] are shown in the last column of Table 1. Our model consistently outperforms other baselines, suggesting that it also contributes to the text segmentation task.

Ablation Models	Sentence-F1	Span-F1
NICTA-PIBOSO		
Ours	**86.8**	**62.9**
- w/o SDLA	85.1	59.7
- w/o supervised signal	84.9	59.1
- w/o span-based classification	85.6	61.0
PubMed 20k RCT		
Ours	**92.8**	**84.5**
- w/o SDLA	92.3	82.4
- w/o supervised signal	92.6	82.9
- w/o span-based classification	92.6	83.4

Table 4: Ablation study on two datasets.

3.4 Ablation Study

To investigate the effectiveness of the designed components, we conduct an ablation study on the proposed model, and the results are listed in Table 4. With the help of the SDLA component, the performances are improved significantly, and the way we impose the supervised signal to guide the attention proves effective for yielding more true positives. And the auxiliary task of span classification effectively improves Span-F1.

3.5 Attention Visualization and Case Study

As shown in Figure 3, by incorporating supervised signal, the attention focus on a continuous local

[1]Please refer to Yamada et al. (2020) for the detailed calculation way of Sentence-F1 and Span-F1.

[2]Since Yamada et al. (2020) don't release their codes, we are unable to evaluate its P_k performance. The P_k results of other models are obtained by running their codes.

Sentence	Gold	Base	Ours
Tizanidine hydrochloride , an alpha (2) - adrenergic receptor agonist , is a widely used medication for the treatment of muscle spasticity .	B	B	B
Clinical studies have supported its use in the management of spasticity caused by multiple sclerosis (MS) , acquired brain injury or spinal cord injury .	B	B	B
It has also been shown to be clinically effective in the management of pain syndromes , such as : myofascial pain , lower back pain and trigeminal neuralgia .	B	B	B
This review summarizes the recent findings on the clinical application of tizanidine .	O	B	O
Our objective was to review and summarize the medical literature regarding the evidence for the usefulness of tizanidine in the management of spasticity and in pain syndromes such as myofascial pain .	O	B	O
We reviewed the current medical and pharmacology literature through various internet literature searches .	O	B	O
This information was then synthesized and presented in paragraph and table form .	O	O	O

Table 5: Examples of label predictions for NICTA-PIBOSO abstract by BERT+BiLSTM+CRF (Base) and our proposed method (Ours). B and O denote background and other labels respectively.

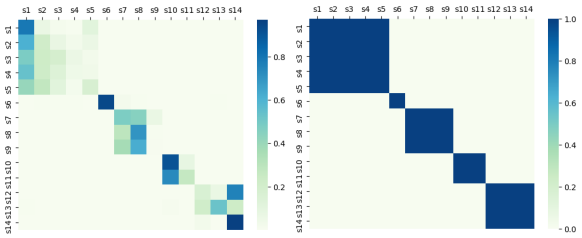


Figure 3: Visualization of attention weights (left) and supervised signal (right). The deeper color means the higher weight.

span around the gold span. The visualization results not only verifies the effectiveness of the supervised signal, but also reveals the interpretability of our proposed SDLA.

Table 5 shows the results of **Base** and **Ours** method for an abstract obtained from NICTA-PIBOSO. Our model correctly identified the boundary between the spans labeled by background (B) and other (O), which shows our model benefit from capturing latent segment structure identifying the more indistinguishable segmentation boundaries.

4 Conclusion

In this paper, we propose a novel model for SSC task, which includes a supervised dynamic local attention to explore the latent segment structure of the document, and an auxiliary task to improve the performance at span-level representations. We demonstrate the effectiveness of our model on two datasets and find that our model also performs well in the text segmentation scenario. In future work, we will consider joint learning sequential sentence classification and text segmentation.

Acknowledgments

We thank the anonymous reviewers for their helpful feedbacks. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant No. 61502174, and 61872148), the Natural Science Foundation of Guangdong Province (Grant No. 2017A030313355, 2019A1515010768 and 2021A1515011496), the Guangzhou Science and Technology Planning Project (Grant No. 201704030051, and 201902010020), the Key R&D Program of Guangdong Province (No. 2018B010107002) and the Fundamental Research Funds for the Central Universities.

References

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Daniel Cohen, Liu Yang, and W Bruce Croft. 2018. Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1165–1168.

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed

200k rct: a dataset for sequential sentence classification in medical abstracts. *arXiv preprint arXiv:1710.06071*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tracy Edinger, Dina Demner-Fushman, Aaron M Cohen, Steven Bedrick, and William Hersh. 2017. Evaluation of clinical text segmentation to facilitate cohort retrieval. In *AMIA Annual Symposium Proceedings*, volume 2017, page 660. American Medical Informatics Association.

Hamed Hassanzadeh, Tudor Groza, and Jane Hunter. 2014. Identifying scientific artefacts in biomedical literature: The evidence based medicine use case. *Journal of biomedical informatics*, 49:159–170.

Kenji Hirohata, Naoaki Okazaki, Sophia Ananiadou, and Mitsuru Ishizuka. 2008. Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Di Jin and Peter Szolovits. 2018. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3100–3109, Brussels, Belgium. Association for Computational Linguistics.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, volume 12, pages 1–10. BioMed Central.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473, New Orleans, Louisiana. Association for Computational Linguistics.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520, San Diego, California. Association for Computational Linguistics.

Jimmy Lin, Damianos Karakos, Dina Demner-Fushman, and Sanjeev Khudanpur. 2006. Generative content models for structural analysis of medical abstracts. In *Proceedings of the hlt-naacl bionlp workshop on linking natural language and biology*, pages 65–72.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2020. Differentiable window for dynamic local attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6589–6599, Online. Association for Computational Linguistics.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2692–2700.

Kosuke Yamada, Tsutomu Hirao, Ryohei Sasano, Koichi Takeda, and Masaaki Nagata. 2020. Sequential span classification with neural semi-Markov CRFs for biomedical abstracts. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 871–877, Online. Association for Computational Linguistics.

Zhi-Xiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-markov crf for neural sequence labeling. *arXiv preprint arXiv:1805.03838*.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, et al. 2020. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3239–3248.

How effective is BERT without word ordering? Implications for language understanding and data privacy

Jack Hessel
Allen Institute for AI
jackh@allenai.org

Alexandra Schofield
Harvey Mudd College
xanda@cs.hmc.edu

Abstract

Ordered word sequences contain the rich structures that define language. However, it's often not clear if or how modern pretrained language models utilize these structures. We show that the token representations and self-attention activations within BERT are surprisingly resilient to shuffling the order of input tokens, and that for several GLUE language understanding tasks, shuffling only minimally degrades performance, e.g., by 4% for QNLI. While bleak from the perspective of language understanding, our results have positive implications for cases where copyright or ethics *necessitates* the consideration of bag-of-words data (vs. full documents). We simulate such a scenario for three sensitive classification tasks, demonstrating minimal performance degradation vs. releasing full language sequences.

1 Introduction

Masked language models (MLMs) like BERT (Devlin et al., 2019) use an ordered sequence of tokens as input. And rightfully so! Any model capable of "language understanding" undoubtedly *should* need access to the hierarchical, syntactic structures implicitly encoded in language. But are MLMs really doing better *because* they have access to full word sequences?

To assess this question, we first compare the internal representations of BERT and RoBERTa (Liu et al., 2019) when the sequence of unigrams is not available.[1] We do this by using the bag-of-words counts of an input to generate a random ordering of the unigrams, i.e., "shuffling" the input. For example, in a sentiment classification corpus, if an intact input was "The movie was great!", a possible shuffled ordering might be "movie the great was" (tokenization details are in §4). We find that, though BERT appears to become more sensitive to ordering in later layers, shuffled token representations and self-attention activations still closely resemble their unshuffled counterparts.

Following cues from prior work (Sugawara et al., 2020; Si et al., 2019; K et al., 2020), we next report the performance of pre-trained MLMs fine-tuned on GLUE, a suite of English-language understanding benchmarks, when given access only to unigram count information by handing models randomly ordered sequences of words (an approach we call BoW-BERT, for short). For most GLUE tasks, performance degradation when shuffling is minimal, e.g., MNLI, QQP, and QNLI accuracy degrade by less than 5 accuracy points.

The bad news: Despite BERT being trained on intact word sequences, BoW-BERT demonstrates that MLMs can readily ignore syntax (while maintaining strong performance) when fine-tuned for even carefully designed downstream language understanding tasks.[2] We thus advocate for reporting BoW-BERT's performance as a strong baseline.

The good news: BoW-BERT offers a practical modeling choice for researchers who *must* operate with only bag-of-words representations for legal or ethical reasons.[3] Bag-of-words data releases are sometimes the *only legal format* in which copyright-sensitive corpora may be distributed, e.g., HathiTrust[4] (16M historical volumes) (Christenson, 2011), Google N-grams (Michel et al., 2011), etc. And while ethical considerations sometimes preclude the full release of privacy-sensitive docu-

[1] We use the "base" models supplied by the authors

[2] Bowman and Dahl (2021) provide perspective on "fixing" NLU tasks.

[3] This is a surprisingly common case: our initial motivation for BoW-BERT was our experience in exploring such a corpus.

[4] In Authors Guild, Inc. v. HathiTrust (2014), the 2nd Circuit U.S. Court of Appeals ruled that showing only "*the number of times [a search term] appears on each page*" constitutes legal fair use, but "[displaying] to the user any text from the underlying copyrighted work" might not.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 204–211
August 1–6, 2021. ©2021 Association for Computational Linguistics

ments (e.g., medical transcriptions), bag-of-words data release offers the potential for compromise. While releasing unigram counts is one way of anonymizing documents (Gallé and Tealdi, 2015), recent work in differential privacy (Dwork, 2008; Fernandes et al., 2019; Schofield et al., 2019; Schein et al., 2019) has resulted in randomized algorithms capable of privatizing BoW count data (under varying definitions of privacy).[5]

We explore classification tasks on three sensitive corpora, simulating different input fidelity availability: full sequences, BoW counts, and differentially private (DP) BoW counts. We find that BoW-BERT often significantly outperforms prior BoW models, especially for shorter documents. And, for longer documents, BoW-BERT can even outperform full-sequence BERT. Finally, for the (naive) DP configuration we consider, BoW-BERT is a viable option for classifying shorter privatized documents, though linear BoW models remain competitive for longer documents.

2 Related Work

Shuffling inputs to non-pretrained models. Word order shuffling has been tested as part of the full training process for non-pretrained models. Sankar et al. (2019) shuffle words in a dialog corpus, and find that LSTMs are more sensitive than Transformers to word order. Khandelwal et al. (2018) show that shuffling distant context words (e.g., beyond 50 tokens) has little effect in outcome for LM-LSTMs. Adi et al. (2017) show that LSTM autoencoders encode significant ordering information when fit to a corpus of Wikipedia sentences. Nie et al. (2019) report minimal performance decreases from word shuffling while training a number of model architectures, e.g., ESIM (Chen et al., 2017), for SNLI/MNLI tasks. In a multimodal setting, Cirik et al. (2018) show that shuffling doesn't affect performance for an LSTM in a referring expression task.

Shuffling inputs to pretrained MLMs. While at the time of submission of this work, shuffling results had not been fully reported on the popular GLUE taskset, prior results have used word-shuffling as a baseline with varying results.

Sugawara et al. (2020) operationalize ablations of reading comprehension skills from Kintsch (1988), and report that shuffling n-grams in 10 QA corpora results in 10-20% performance decreases for BERT. Si et al. (2019) report similar results when shuffling questions+answers in MCRC corpora, reporting absolute accuracy drops of between 5-20% when shuffling both passage/question words (e.g., BERT on DREAM drops from 63 → 41 accuracy relative to a 33% constant baseline). K et al. (2020) report that swapping tokens during pretraining of a multilingual BERT model results in moderate performance degradation for XNLI (e.g., 71 → 63 for en-es) but more significant performance degradation for NER (63 → 40 in the same setting). They find that a purely frequency-based corpus "is not enough for a reasonable cross-lingual performance."

Several works have examined shuffling inputs in multi-language scenarios (e.g., translation) when languages have variable syntax (Ahmad et al., 2019; Liu et al., 2020). Zhao et al. (2020) use a random token permutation to provide a baseline. Yang et al. (2019) find that self-attention networks are surprisingly bad at identifying two tokens that are swapped in the input. Ettinger (2020) show that shuffling BERT inputs decreases word cloze prediction performance on a corpus of 102 sentences without fine-tuning. Wang et al. (2020) incorporate a deshuffling objective into pre-training.

In some cases, shuffled inputs provide a stronger baseline than might be assumed, while in others, shuffling significantly degrades performance. At present, determining whether or not order is "needed" for a particular task is largely an experimental, empirical endeavor.

Syntax in MLMs. Prior works have investigated BERT's capacity to represent syntax: some researchers have designed prediction tasks that require syntactic knowledge (Linzen et al., 2016; Jawahar et al., 2019; Lin et al., 2019; Goldberg, 2019), while others have probed representations for linguistic information directly (Mareček and Rosa, 2018; Liu et al.; Hewitt and Manning, 2019; Reif et al., 2019). Tenney et al. (2019) find that contextual representations outperform lexical representations on many syntactic tasks, but not in a suite of semantic prediction tasks. Htut et al. (2019) and Clark et al. (2019) find that some attention heads encode information useful for dependency parsing. Glavaš and Vulić (2020) show that intermediate supervised training of a biaffine parser has little effect on downstream MLM performance.

[5] Releasing BoW counts is related to, but distinct from, the setting considered by Beigi et al. (2019), who produce private vector representations with uninterpretable dimensions.

A Bouquet of Contemporaneous Work. While this work was in submission, several related works were posted to arXiv. Gupta et al. (2021) examine NLI, paraphrase detection, and sentiment classification, and show that destructive interventions do not significantly affect either model predictions or model confidence. Sinha et al. (2020) find a similar result for NLI tasks, and, in follow-up work, Sinha et al. (2021) demonstrate pretraining is possible on unordered sequences. Pham et al. (2020) look specifically at GLUE classification for BERT-based models. Beyond contemporaneous confirmation of the GLUE results, our work contributes to this bouquet by: 1) examining internal activations/layers and 2) exploring classification settings where one might need to operate on (potentially differentially private) count-only data.

3 Representation analysis

We might expect that shuffling the order of tokens in an input sentence would significantly corrupt the internal representations of BERT, but is that actually the case? We investigate with two new metrics. Consider applying a pre-trained, fixed BERT model to $x =$"the movie was great" and the shuffled $x' =$"movie the great was".

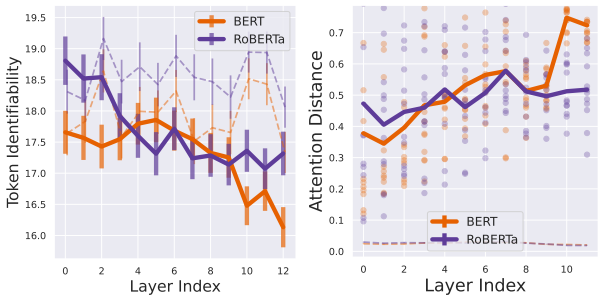
Token identifiability measures the similarity of BERT's vector representations of a word token (e.g., "movie") in x and x'. Identifiability is high if the model has similar representations for tokens after their order is shuffled.

Self-attention distance measures if BERT attends to similar tokens for each token in x and x' regardless of their order (e.g., is "the movie was great" $\approx$ "movie the great was" to BERT?). Self-attention distance is low if the model attends to the same tokens after input shuffling.

Token Identifiability. Let $\texttt{MLM}_l(x)$ be a $\mathbb{R}^{t\times d}$ matrix, where t is the number of tokens in sentence x, d is the MLM's dimension, and l is the layer index. In this setting, row i of $\texttt{MLM}_l(x)$ is the MLM's representation of the ith token in sentence x. We compare $\texttt{MLM}_l(x)$ to $\mathbb{E}[\texttt{MLM}_l(X')]$, where X' is drawn uniformly from the permutations of x: $perm(x)$. For a specific sample $x' \sim perm(x)$, we first take the row-wise cosine similarity of $\texttt{MLM}_l(x)$ and $\texttt{MLM}_l(x')$, and treat the resulting $t \times t$ matrix as an instance of a bipartite linear assignment problem. The *assignment accuracy* (AA) score for (x, x') is the proportion of assigned token pairs that have the same underlying word type. To avoid biasing towards shorter sentences, we take the ratio of the accuracy relative to chance, i.e.,

$$\text{ID-MLM}(x, l) = \frac{\mathbb{E}_{X'}[\text{AA}(\texttt{MLM}_l(x), \texttt{MLM}_l(X'))]}{\mathbb{E}_{\texttt{RAND}}[\text{AA}(\texttt{MLM}_l(x), \texttt{RAND})]}, \quad (1)$$

where RAND is a random matrix of reals $\mathbb{R}^{t\times d}$.[6]



(a) Identifiability (b) Self-Attention Distance

Figure 1: Token identifiability and attention distance by layer for BERT and RoBERTa; dashed lines represent baseline values of metrics with unshuffled sequences, error bars are 95% CI for mean, scatterplot=per-attention head result. Identifiability decreases towards 1 (pure random token features) when shuffled inputs produce very different embeddings from the intact inputs, while self-attention distance increases towards 1 (pure random attention) in this case. While later layers in both models are more order-sensitive, information is retained for shuffled inputs.

Self-Attention Distance. Let $\texttt{AMLM}_{l,h}(x)$ be the row-l_1-normalized $\mathbb{R}^{t\times t}$ matrix representing the self-attention matrix at layer l for attention head h. We can compute the same matrix for a shuffled input $\texttt{AMLM}_{l,h}(x')$, and then perform a transformation to re-order the rows and columns of this matrix to match the original order of tokens in x, yielding $\texttt{AMLM}^x_{l,h}(x')$. We then define the *row-wise Jensen-Shannon divergence* DS-JSD$(\texttt{AMLM}_{l,h}(x), \texttt{AMLM}^x_{l,h}(x'))$ as the mean row-wise JSD between $\texttt{AMLM}_{l,h}(x)$ and the DeShuffled reordered attention matrix $\texttt{AMLM}^x_{l,h}(x')$. As before, to reduce the effect of sentence length, we normalize using RND-JSD$(\texttt{AMLM}_{l,h}(x), \texttt{AMLM}^x_{l,h}(x'))$, which chooses a random row/column permutation. [7] The

[6] In practice, we simply compute the assignment step of AA using a $\mathbb{R}^{t\times t}$ matrix drawn from $U[0, 1)$.

[7] If there are multiple possible valid permutations of x' that match x (e.g., if there are repeated words), DS-JSD will choose the order that minimizes the JSD, and RND-JSD will search through a number of random orderings equal to the number of valid permutations. If the number of valid permutations is > 16, 16 random valid permutations are sampled.

	MNLI-(m/mm) Acc/Acc	QQP F1/Acc	QNLI Acc	SST-2 Acc	CoLA MCC	STS-B PCC-r/SCC-ρ	MRPC F1/Acc	RTE Acc
RoBERTa (full seq)	87.3/87.1	72.0/88.8	92.9	95.8	58.8	89.5/88.8	90.2/86.6	69.9
BoW-RoBERTa	81.1/82.8	68.8/87.5	86.8	85.5	10.4	85.0/83.8	82.1/76.6	58.8
BERT (full seq)	84.2/83.2	71.6/89.1	90.6	92.6	50.7	87.3/86.4	87.5/82.8	68.4
BoW-BERT	79.8/79.7	68.3/87.5	86.2	86.7	14.3	81.8/80.3	82.9/75.2	60.4
CBOW GloVe	56.0/56.4	51.4/79.1	72.1	80.0	0.0	61.2/58.7	81.5/73.4	54.1

Table 1: GLUE test set prediction results.

final *attention distance* metric is defined as

$$\text{AD-MLM}(x, l, h) = \frac{\mathbb{E}_{X'}[\text{DS-JSD}(\texttt{AMLM}_{l,h}(x), \texttt{AMLM}^{x}_{l,h}(X'))]}{\mathbb{E}_{X'}[\text{RND-JSD}(\texttt{AMLM}_{l,h}(x), \texttt{AMLM}^{x}_{l,h}(X'))]} \quad (2)$$

Results. We randomly sample 100 sentences from each training set of 8 GLUE tasks, for a total of 800 sentences. To approximate expectations from Equations 1 and 2, we sample 32 random permutations per sentence. Figure 1 gives the per-layer token identifiability/attention similarity scores for both MLMs. For both metrics, later layers are more order sensitive to order, i.e., ID-MLM ↓ and AD-MLM ↑. Attention heads vary significantly in their order sensitivity: each attention head is a single point in the scatterplot of Figure 1b. But, even at late layers, both metrics suggest significantly more than random correspondence: internal representations of BoW-(Ro)BERT(a) clearly resemble their unshuffled counterparts.

4 BoW-BERT for Classification

We compare BERT and RoBERTa to their BoW counterparts on nine tasks from GLUE (Wang et al., 2019).[8] We run single-task training for six epochs, use early stopping, and optimize batch size ($\{16, 32\}$) and learning rate ($\{5, 2, 1, .5\} \times 10^{-5}$) via grid search on the validation set. To shuffle documents: we lowercase, tokenize, remove all tokens that consist only of punctuation, shuffle, then concatenate with whitespaces. We re-shuffle the training tokens each epoch, but fix validation and test tokens to one shuffled permutation.

[8]These tasks span NLI (MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), and RTE (Dagan et al., 2006; Bar Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009)); semantic similarity estimation (QQP,[9] MRPC (Dolan and Brockett, 2005), STS-B (Cer et al., 2017)); sentiment analysis (SST-2 (Socher et al., 2013)), and grammaticality judgement (CoLA (Warstadt et al., 2019)). We omit WNLI (Levesque et al., 2011) as is common (all models achieve chance performance on that corpus).

Results. Table 1 gives the GLUE test set results of our algorithms vs. GloVe CBOW, the best BoW baseline on the GLUE leaderboard at the time of submission. In all cases BoW-BERT outperforms CBOW. The extent to which BoW-BERT underperforms relative to BERT varies for each dataset, but in terms of relative percent performance decrease, ranges from over ↓70% for CoLA to only ↓3% QQP. Outside of CoLA, performance degradation never exceeds 10 absolute points for any task's metric.

According to the GLUE diagnostic set (which tests 33 categories of linguistic phenomena) BoW-BERT has the most trouble with dealing with double negations (e.g., "I have never seen a hummingbird not flying.": MCC degrades 31.7 → -4.3 when switching BERT → BoW-BERT), quantifiers ("our sympathy to all [vs. some] of the victims": 61.8 → 46.1); and temporal logic ("Mary left before John entered": 8.0 → -8.6). Results for GLUE diagnostic meta-categories are: Knowledge (24.4 → 24.3); Pred-Arg Structure (39.2 → 39.1); Logic (24.7 → 22.1); Lexical Semantics (39.7 → 31.5).

Classification for Sensitive Texts

Privacy and legal concerns frequently necessitate BoW-only data releases. We ask: for potentially sensitive text classification tasks, *how does performance degrade if only bag of words counts are available (instead of full sequences)?* We consider three such tasks: Reddit controversy prediction on AskWomen/AskMen (CONT) (Hessel and Lee, 2019), offensiveness prediction in social media (SBF) (Sap et al., 2020), and sample medical transcript categorization (MTSAMP).[10] For each task, we compare models with access to sequences vs. models that can only access bag-of-words features. Our baselines are unigram/tfidf linear models, and CBOW models GloVe and fasttext (Mikolov et al., 2018). Table 2 contains corpus

[10]https://www.mtsamples.com/

statistics and prediction results. For CONT and SBF, BoW-BERT outperforms all BoW methods. For all tasks, performance drop-off from a full-sequence fine-tuned MLM to its BoW counterpart is less than 1%. CBOW/tfidf remain strong for MTSAMP, in which documents are longer.

Given that de-shuffling BoW representations is at least partially possible (Tao et al., 2021), we additionally consider a more robust *differentially private* (DP) unigram count data release (also known as the "local model" of DP) (Warner, 1965; Dwork et al., 2006; Schein et al., 2019). We follow a process similar to Schofield et al. (2019) by first compressing the original unigram count matrices via Gaussian random projection to 500D.[11] In the compressed space, we add noise per-entry with the Laplace mechanism (Dwork et al., 2006) with a per-feature privacy budget of ε. Then, we invert the random projection, normalize the vector to be a categorical word distribution, and sample (unordered) pseudodocuments from the resulting distribution with length $\sim Poisson(\ell)$.

We report results in an easier setting $\ell = 256, \varepsilon = 100$ and a harder setting $\ell = 128, \varepsilon = 50$ in the bottom half of Table 2. For these settings of DP, the linear baselines generally outperform BoW-(Ro)BERT(a). However, MLMs are again most competitive for the shortest document setting, SBF, where BoW-(Ro)BERT(a) exceeds the best linear model performance (60.4 vs. 62.0 F1).

Taken together, these results suggest 1) that releasing word counts instead of full document sequences is a viable data release strategy for some sensitive classification tasks; 2) BoW-BERT offers a means of accessing the representational power of modern MLMs in cases where only BoW information is available; and 3) for at least some local DP settings, linear models remain competitive particularly for long documents, while BoW-RoBERTa is viable when the underlying documents are shorter.

5 Conclusion and Future Work

We advocate for BoW-(Ro)BERT(a) as a surprisingly strong baseline for language understanding tasks, as well as a performant practical option for classifying (privatized) BoW texts when documents are short. Future work includes:

1. Evaluating BoW-BERT representations on BoW-only corpora in unsupervised text clustering scenarios (vs. classification) + designing self-supervised objectives for fine-tuning MLM weights from unlabelled domain-specific BoW corpora, e.g., HathiTrust.;
2. Extending (K et al., 2020) by further exploring BoW classification using non-English MLMs, where model dependence on syntactic information may differ;
3. Designing local private data release methods better adapted to MLM fine-tuning.

Acknowledgments. We thank David Mimno, Gregory Yauney, Chandra Bhagavatula, Keisuke Sakaguchi, and Max Chen for helpful discussions, comments, suggestions, and (in one case) physically rescuing files from an unplugged server. We also thank Gautam Kamath for feedback on an earlier version of the differential privacy section. Finally, we thank both our EACL and ACL reviewers for their thoughtful feedback.

	CONT	SBF	MTSAMP
Mean len (toks)	111	23	578
# of docs	6.3K	45K	5.0K
# classes	2	2	40
	Acc	F1	Acc/W-F1
BERT (full seq)	65.2	84.1	30.1/27.4
BoW-BERT	**64.1**	**83.4**	34.3/29.6
RoBERTa (full seq)	*66.5*	*84.8*	31.5/29.1
BoW-RoBERTa	62.9	82.9	34.9/32.0
CBOW fasttext	61.7	77.7	***39.4/36.0***
CBOW GloVe	61.1	77.0	38.8/35.2
Unigram tfidf	57.3	78.9	36.2/25.0
Unigram Counts	58.0	79.5	33.5/20.6
Popular Class	50.0	0.0	20.7/7.1
Random Prediction	51.2	47.3	8.9/8.5
$DP^{\varepsilon=100}_{\ell=256}$ BoW-BERT	53.4	59.5	29.0/15.7
$DP^{\varepsilon=100}_{\ell=256}$ BoW-RoBERTa	53.0	62.0	28.9/14.9
$DP^{\varepsilon=100}_{\ell=256}$ Best Linear	57.7	60.4	31.3/21.5
$DP^{\varepsilon=50}_{\ell=128}$ BoW-BERT	50.5	57.0	22.4/10.8
$DP^{\varepsilon=50}_{\ell=128}$ BoW-RoBERTa	51.8	58.9	21.8/10.7
$DP^{\varepsilon=50}_{\ell=128}$ Best Linear	55.0	58.8	25.9/17.8

Table 2: Top: text classification prediction results on sensitive texts; **best BoW** bolded, *best overall* italicized. Bottom: *DP* = results on differentially private data; "Best Linear" is the most performant linear model, tfidf for $DP^{\varepsilon=100}_{\ell=256}$ and unigram counts for $DP^{\varepsilon=50}_{\ell=128}$.

[11] Our original submission used DP PCA instead. But it was brought to our attention that the paper proposing that algorithm was retracted for being non-private (+ discontinued in the library we used after we submitted). We have adjusted our code and recompiled our experiments using a comparable mechanism. Our intent isn't to advocate for this particular DP method, but rather, to fairly compare NLP algorithms on the same DP corpora.

References

Authors Guild, Inc. v. Hathitrust, 755 F.3d 87 (2d Cir. 2014).

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *ICLR*.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *NAACL*.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL recognising textual entailment challenge.

Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. 2019. I am not what I write: Privacy preserving text representation learning. *arXiv preprint arXiv:1907.03189*.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge.

Samuel R Bowman and George E Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In *NAACL*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *ACL*.

Heather Christenson. 2011. Hathitrust. *Library Resources & Technical Services*, 55(2):93–102.

Volkan Cirik, Louis-Philippe Morency, and Taylor Berg-Kirkpatrick. 2018. Visual referring expression recognition: What do systems actually learn? In *NAACL*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*.

Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference*, pages 265–284. Springer.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *TACL*.

Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. In *ETAPS*. Springer-VDI-Verlag GmbH & Co. KG.

Matthias Gallé and Matías Tealdi. 2015. Reconstructing textual documents from n-grams. In *KDD*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*.

Goran Glavaš and Ivan Vulić. 2020. Is supervised syntactic parsing beneficial for language understanding? an empirical investigation. *arXiv preprint arXiv:2008.06788*.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. BERT & family eat word salad: Experiments with text understanding. *arXiv preprint arXiv:2101.03453*.

Jack Hessel and Lillian Lee. 2019. Something's brewing! early prediction of controversy-causing posts from discussion features. In *NAACL*, Minneapolis, Minnesota.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL*.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in BERT track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *ACL*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *ICLR*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *ACL*.

Walter Kintsch. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological review*, 95(2):163.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *TACL*, 4:521–535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *NAACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zihan Liu, Genta Indra Winata, Samuel Cahyawijaya, Andrea Madotto, Zhaojiang Lin, and Pascale Fung. 2020. On the importance of word order information in cross-lingual sequence labeling. *arXiv preprint arXiv:2001.11164*.

David Mareček and Rudolf Rosa. 2018. Extracting syntactic trees from transformer encoder self-attentions. In *Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A Nowak, and Erez Liberman Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of NLI models. In *AAAI*.

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *arXiv preprint arXiv:2012.15180*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *NeurIPS*.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? An Empirical Study. In *ACL*.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *ACL*.

Aaron Schein, Zhiwei Steven Wu, Alexandra Schofield, Mingyuan Zhou, and Hanna Wallach. 2019. Locally private Bayesian inference for count models. In *ICML*.

Alexandra Schofield, Gregory Yauney, and David Mimno. 2019. Combatting the challenges of local privacy for distributional semantics with compression. In *PriML Workshop at NeurIPS*.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does BERT learn from multiple-choice reading comprehension datasets? *arXiv preprint arXiv:1910.12391*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2020. Unnatural language inference. *arXiv preprint arXiv:2101.00010*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *AAAI*.

Chongyang Tao, Shen Gao, Juntao Li, Yansong Feng, Dongyan Zhao, and Rui Yan. 2021. Learning to organize a bag of words into sentences with neural networks: An empirical study. In *NAACL*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. StructBERT: Incorporating language structures into pretraining for deep language understanding. In *ICLR*.

Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *TACL*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. Assessing the ability of self-attention networks to learn word order. In *ACL*.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *ACL*.

WikiSum: Coherent Summarization Dataset for Efficient Human-Evaluation

Nachshon Cohen[*]
Amazon
nachshonc@gmail.com

Oren Kalinsky[*]
Amazon
orenk@amazon.com

Yftah Ziser[*]
Facebook[†]
yftahz@fb.com

Alessandro Moschitti
Amazon
amosch@amazon.com

Abstract

Recent works have made significant advances on summarization tasks, facilitated by summarization datasets. Several existing datasets have the form of coherent-paragraph summaries. However, these datasets were curated from academic documents written for experts, making the essential step of assessing the summarization output through human-evaluation very demanding.

To overcome these limitations, we present a dataset[1] based on article summaries appearing on the WikiHow website, composed of how-to articles and coherent-paragraph summaries written in plain language. We compare our dataset attributes to existing ones, including readability and world-knowledge, showing our dataset makes human evaluation significantly more manageable and effective. A human evaluation conducted on PubMed and the proposed dataset reinforces our findings.

1 Introduction

Summarization is the task of preserving the key information in a text while reducing its length. Recently, many summarization datasets were published and helped push the boundaries of new summarization systems. These datasets differ on several properties, including the domain (e.g., academic or news) and the summary form. PubMed, arXiv, and BigPatent (Cohan et al., 2018; Sharma et al., 2019) provide a summary in the form of coherent paragraphs (i.e., each sentence flows smoothly into the next). In contrast, other summarization datasets (Hermann et al., 2015; Grusky et al., 2018; Koupaee and Wang, 2018; Ladhak et al., 2020) offer a summary in the form of a key points list (i.e., highlights). In this paper, we focus on coherent paragraph summarization datasets.

How to Bake Chicken Breast? To bake chicken breast, start by lining a baking dish with foil or parchment paper. Then, put the chicken in the baking dish and bake it for 30-40 minutes at 400 degrees Fahrenheit, or until it reaches an internal temperature of 160 degrees Fahrenheit.
How to Break Up with Your Friend? The best way to break up with a friend is to confront them. Choose a time and place to meet up and explain to them why you are ending the friendship. Allow your friend to speak their mind as well, and work together to set boundaries for moving forward.

Figure 1: Examples of how-to questions and their corresponding answer's summarization in WikiSum.

Automatic evaluation of summarization systems, e.g., by using the ROUGE metric, is challenging (Lloret et al., 2018) and is often inconsistent with human evaluation (Liu and Liu, 2008; Cohan and Goharian, 2016; Tay et al., 2019; Huang et al., 2020). To understand – and later improve – the quality of summarization systems, it is necessary to conduct a human evaluation. A human evaluation's quality depends on the ease of reading and understanding of the measured text: a simple text does not require annotators with unique expertise, can be evaluated faster, and is easier to annotate correctly. However, existing coherent-paragraph summarization datasets consist of academic papers and cannot be considered easy to read. Evaluating such summarization samples requires unique expertise, takes time, and comes at a high cost.

In this work, we present WikiSum, a new summarization dataset from the WikiHow knowledge base[2]. The WikiSum documents are written in simple English, and the summaries provide "non-obvious tips that mimic the advice a knowledgeable, empathetic friend might give."[3] Unlike previous WikiHow summarization (Koupaee and Wang, 2018; Ladhak et al., 2020) datasets and summaries

[*] Co-first author
[†] Work done while at Amazon
[1] The dataset and human evaluation are available at https://registry.opendata.aws/wikisum.
[2] https://www.wikihow.com
[3] https://www.wikihow.com/Write-or-Edit-a-Quick-Summary-on-wikiHow

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 212–219
August 1–6, 2021. ©2021 Association for Computational Linguistics

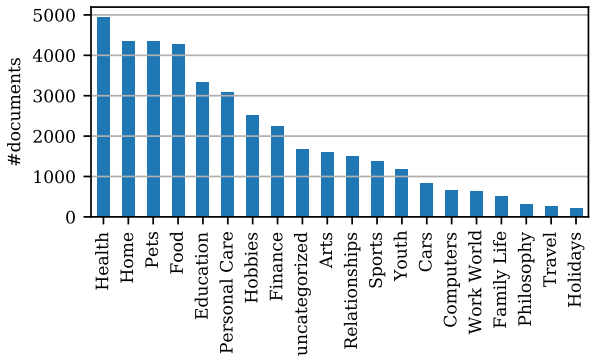


Figure 2: Category distribution in WikiSum.

from the news domain, the summaries of WikiSum are in the form of a coherent paragraph written by the document authors (examples in Figure 1). Moreover, in contrast to other coherent-paragraph summarization datasets from the academic domain, WikiSum is written using simple English. This critical property can help with the challenging task of evaluating summarization systems and provide insights that can go unnoticed using automatic evaluation methods.

The key attributes of WikiSum are: (1) Summaries written as a single, coherent passage. (2) Articles and summaries that are easy to read. (3) Articles and summaries require less world knowledge to understand. We evaluate the dataset readability and estimate the required world-knowledge in Section 3. Moreover, we reinforce our results by conducting a human-evaluation of a summarization dataset in Section 4. Finally, to establish a baseline on the proposed dataset, we benchmark WikiSum using recent summarization systems and report their performance on Section 5.

2 Related Work

The summarization landscape can be roughly divided into three primary summary-forms: (1) Single sentence (Napoles et al., 2012; Grusky et al., 2018; Narayan et al., 2018; Kim et al., 2019) - summarize the document in a single sentence; (2) Highlights (Hermann et al., 2015; Koupaee and Wang, 2018; Ladhak et al., 2020) - a summary in the form of bullets listing the key points in the text; (3) Coherent summary (Sharma et al., 2019; Cohan et al., 2018) - short coherent paragraphs describing the salient information. The summarization datasets from the news domain, which are commonly used for human evaluation, include summaries in the form of highlights or single-sentence summaries. However, summarization datasets written in a coherent format come from the academic domain, making them extremely difficult to annotate manually. Our proposed WikiSum is the only dataset written in a coherent format, yet easy for human evaluation. We do not claim that coherent paragraph summaries are *better*, but rather *different*; each format has its use cases, and human evaluation should be done on each of the different formats separately.

The existing WikiHow datasets (Koupaee and Wang, 2018; Ladhak et al., 2020) can be considered the closest to WikiSum, as they originate from the same knowledge base. However, while the existing WikiHow datasets split the article to generate the document and summary, WikiSum uses the entire article as the document and a summary specifically written by the article's author (called the Article Quick Summary). The former uses the concatenation of the first line of each step, called the step header, as the list of highlights and the remainder of step text's concatenation called "wrap-text," as the document[4]. In addition to the different summary-form of the highlight-based WikiHow and WikiSum, the content of the summaries is significantly different, which can be illustrated by the low BLEU-4 (0.06[5]) between the two.

BigPatent (Sharma et al., 2019), Arxiv and PubMed (Cohan et al., 2018) are recent summarization datasets with coherent paragraph summaries. These datasets focus on the academic domain and are written for experts. Like these datasets, WikiSum is composed of long documents and coherent paragraph summaries. Nonetheless, it uses common everyday language and ranges over many domains (see Figure 2). Finally, Table 1 compares WikiSum to common existing datasets. Additional details on WikiSum are available in the appendix.

3 Measuring Text Difficulty

This section focuses on two crucial attributes: ease of readability and external knowledge required, shown (in Section 4) to be important for easy and effective human evaluation. For brevity, we focus on summarization datasets with coherent-paragraph summaries.

[4]WikiHow author instructions (wikihow.com, 2020) specifically states that the authors can use the wrap-text to describe why the step header is important. This leads to many cases where the step headers are not a summary of the wrap-text.

[5]We used WikiSum as the reference, the results are very similar when WikiHow is used as a reference. ROUGE-1, 2 and L are 0.37, 0.13, and 0.23, respectively.

	Domain	# Docs	Comp. ratio	Summary # word	Summary # sent	Doc # word
WIKISUM	instructional	39,775	13.9	101.2	5.0	1,334.2
ARXIV	academic	215,913	39.8	292.8	9.6	6,913.8
PUBMED	academic	133,215	16.2	214.4	6.9	3,224.4
BIGPATENT	academic	1,341,362	36.4	116.5	3.5	3,572.8
WIKIHOW	instructional	215,365	14.5	69.0	7.2	500.8
CNN/DM	news	312,085	13.0	55.6	3.8	789.9
NYT	news	654,788	12.0	44.9	2.0	795.9
NEWSROOM	news	1,212,726	43.0	30.4	1.4	750.9
XSUM	news	226,711	18.8	23.3	1.0	431.1

Table 1: Statistics comparison of summarization datasets. Datasets not in coherent-paragraph form are marked in gray.

3.1 Readability

Readability metrics attempt to indicate how difficult a passage in English is to read. We used classical readability measures, including FKGL (Farr et al., 1951), GFI (Robert, 1968), SMOG (Mc Laughlin, 1969), ARI (Senter and Smith, 1967), CLI (Coleman and Liau, 1975). All these metrics are based on lexical features of the text, e.g., number of words in a sentence or mean number of syllables per word. They produce a score that is interpreted as the number of years of formal education required (for a native English speaker) to understand a piece of text[6].

For each document, we measured readability scores[7] for the document and the ground truth summary. The document is longer than the summary, so its readability is of higher importance. We report the average readability score for all the samples in the dataset.

Readability scores for the documents are presented at the top of Table 2. The table shows that WikiSum is significantly easier to read than other documents from coherent-summary datasets (arXiv, PubMed, BigPatent). Similar results can be found for the readability scores for the summaries (bottom of Table 2). To conclude, WikiSum is measured as drastically simpler to read than other coherent-summary datasets.

3.2 External Knowledge

Existing datasets are composed of academic documents that are written for experts. Often, to fully understand academic texts requires domain knowledge, which makes the annotator pool smaller, and thus, in most cases, more expensive. Word frequency is a strong indicator of how familiar a word is (Paetzold and Specia, 2016), where rare words tend to be less familiar.

We used OpenSubtitles (Lison and Tiedemann, 2016), text corpora compiled from an extensive database of movie and TV subtitles to obtain word frequencies. We hypothesize that movie and TV subtitles can roughly represent common knowledge among many people. In Figure 3, we show the percentage of non-frequent words in a document (i.e., words that cannot be found in the top-k words in OpenSubtitles) as a function K, averaged over a random sample of $10,000$ documents from each dataset. This figure clearly shows that WikiSum is composed of significantly fewer words unpopular in TV shows and movies, requiring less specialized external knowledge.

[6]Other readability metrics such as FRE (Flesch, 1948), LIX and RIX (Björnsson, 1968), have a similar trend to the shown metrics, but require a translation to years of education, omitted from this paper for brevity.

[7]https://github.com/mmautner/readability

	Dataset	ARI	FKGL	GFI	SMOG	CLI
Document	WikiSum	7.4	6.82	10.15	9.71	8.83
	arXiv	14.02	13.51	18.47	15.44	14.31
	PubMed	16.74	16.27	20.64	17.03	15.01
	BigPatent	13.46	13.32	17.47	14.68	11.68
Summary	WikiSum	9.71	8.49	11.91	10.24	8.78
	arXiv	16.44	16.1	20.5	16.8	15.23
	PubMed	17.73	17.35	21.6	17.44	16.6
	BigPatent	22.47	20.91	25.12	18.75	14.0

Table 2: Readability scores for the documents (top) and summaries (bottom), measured in years of formal education required to read the text. Smaller is simpler.

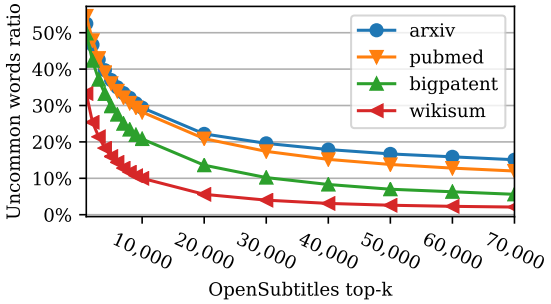


Figure 3: Ratio of uncommon words in the document, which cannot be found in the Top-K OpenSubtitles words, for different k values.

4 Human Evaluation

We conducted a standard human evaluation on a summarization task, in addition to the automatic readability and the external knowledge metrics. We gathered a pool of 6 annotators, without any prior knowledge of the project, all with a graduate degree (M.sc. or Ph.D.) and proficient English reading-level. We asked them to evaluate summaries generated by Pegasus (Zhang et al., 2020). The an-

dataset	time (minutes)	difficulty (rating)	exhausting (rating)	qualified (rating)	unknown (%)
WikiSum	6.8±1.2	1.9±0.3	2.2±0.5	4.2±0.3	0.2±0.1
PubMed	10.0±1.2	3.7±0.3	3.9±0.4	2.2±0.4	3.7±1.4

Table 3: Evaluation time per sample, evaluation difficulty/exhaustion rating, perceived qualification, and the ratio of unknown words in the document. ± denotes 95% confidence interval according to student's t distribution (df=20). Difficulty, qualification, and tiring were marked on a 1-5 scale.

notation task followed Huang et al. (2020) and consisted of relevance, consistency, fluency, and coherency.

Due to resource limitations (and the difficulty of annotating articles from the academic domain), we had to pick one coherent-paragraph dataset for comparison with WikiSum. To avoid annotators' domain bias, we selected articles from PubMed, which contains articles not in the area of expertise of any annotator, in addition to WikiSum. We sampled random articles with 950 - 1050 words to avoid length bias, ensuring that article length is similar in both datasets. All annotators allocated 1 hour, which amounted to 42 annotations, 21 for each dataset.

During the annotation task, we measured the evaluation time and asked the annotators to mark unfamiliar words. In addition, we asked the annotators to rate the following aspects on a 1-5 scale: (a) How difficult was the task? (b) How tiring was it? (c) How qualified are you for this task? After each pair of PubMed and WikiHow samples were completed, the annotators selected which dataset they prefer to evaluate.

In Table 3 we show the annotators' assessment of the tasks. Compared to PubMed, a WikiSum annotation takes significantly less time, is less difficult, and less tiring. Moreover, the annotators revealed that they were much more qualified to assess the WikiSum task summary. Finally, in 90% of the cases (19 out of 21), the annotators revealed that they preferred a WikiSum annotation task. This reinforces our findings that WikiSum is significantly easier to annotate than PubMed.

In the annotation task, we also asked the annotators to mark unfamiliar words in the article. We found a strong correlation between the count of unfamiliar words and the task difficulty, evaluation time, and perceived required qualification (Pearson correlation of $0.57, 0.36, -0.48$[8], respectively, $p < 0.05$). Strong correlation was also found between the ARI readability metric (Section 3.1) and the above-mentioned annotation metrics (Pearson correlation of $0.69, 0.49, -0.76$, $p < 0.05$). This demonstrates the effect of readability on the difficulty of an annotation task.

Finally, we found that unfamiliar words correspond to low-frequency OpenSubtitles words (Section 3.2). The *unfamiliar* words on WikiSum and PubMed appear in the top $91,550$ and $230,596$ words on average, respectively, while *familiar* words appear in the top $16,935$ and $59,244$ words on average, respectively. It also further validates Paetzold and Specia (2016) hypothesis about the strong correlation between word frequency and complexity.

Models	LEAD-3	TextRank	PEGASUS$_{\text{LARGE}}$
WIKISUM	25.3/6.84/16.2	32.7/8.8/18.9	43.35/15.48/26.91
ARXIV	25.53/5.98/15.22	33.1/9.7/18.1	43.07/19.70/34.79
PUBMED	26.38/8.73/16.6	35.3/13.1/20.4	44.70/17.27/25.80
BIGPATENT	28.9/7.96/18.17	33.0/9.8/19.6	45.49/19.90/27.69

Table 4: ROUGE-1/2/L F1 scores on coherent-summary datasets. Pegasus baseline results are from (Zhang et al., 2020), except for WikiSum.

5 Model Results and Discussion

To provide both abstractive and extractive baselines for WikiSum, we evaluate on PEGASUS$_{\text{LARGE}}$ (Zhang et al., 2020), TextRank (Mihalcea and Tarau, 2004), and the common LEAD-3 that selects the first three sentences of the document as the summary. We compare the results on WikiSum to the Arxiv, PubMed, and BigPatent Datasets results. Table 4 reports the F1 scores of ROUGE-1, 2 and L for all the models. The results show that the models' performance on WikiSum is not drastically different from the other datasets, making it an interesting dataset for benchmarking summarization systems. The detailed evaluation setup can be found in the supplementary materials.

To conclude, this paper presents the WikiSum dataset, which is drastically simpler for human evaluation than existing summarization datasets where the summary appears as a coherent paragraph. We showed WikiSum's simplicity via various readability metrics and demonstrated that the text requires less external knowledge to be understood. Finally, we validated our finding via a human evaluation task on WikiSum and PubMed.

[8]Many unfamiliar words implied annotators perceived themselves as less unqualified.

References

Carl Hugo Björnsson. 1968. Läsbarhet, stockholm: Liber.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 615–621. Association for Computational Linguistics.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. *CoRR*, abs/1604.00400.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

James N Farr, James J Jenkins, and Donald G Paterson. 1951. Simplification of flesch reading ease formula. *Journal of applied psychology*, 35(5):333.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 708–719. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2519–2531. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen R. McKeown. 2020. Wikilingua: A new benchmark dataset for multilingual abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 4034–4048. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Feifan Liu and Yang Liu. 2008. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio. Association for Computational Linguistics.

Elena Lloret, Laura Plaza, and Ahmet Aker. 2018. The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52(1):101–148.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411. ACL.

Courtney Napoles, Matthew R. Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, AKBC-WEKEX@NAACL-HLT 2012, Montrèal, Canada, June 7-8, 2012*, pages 95–100. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Gunning Robert. 1968. The technique of clear writing. *Revised Edition). New York: McGraw Hill*.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, CINCINNATI UNIV OH.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.

Wenyi Tay, Aditya Joshi, Xiuzhen Jenny Zhang, Sarvnaz Karimi, and Stephen Wan. 2019. Red-faced rouge: Examining the suitability of rouge for opinion summary evaluation. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 52–60.

wikihow.com. 2020. WikiHow Article Guidelines. https://www.wikihow.com/Write-a-New-Article-on-wikiHow.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

A Data Description

A.1 Gathering the data

We use Scrapy scraper[9] to download articles and summaries from the wikihow.com website. We removed HTML tags using BeautifulSoup[10]. Finally, we removed any sample in which the summary is a list of bullet points; around 7k samples were excluded in this manner.

A.2 Authors Instructions for Writing Quick Summaries

The wikihow.com website provides the following guidelines for authors writing a quick summary.[11]

> The goal of the "Quick Summary" section on wikiHow is to provide a short summary of non-obvious tips that mimic the advice a knowledgeable, empathetic friend might give you if you asked them for help on the given topic. Among other uses, Quick Summaries help smart devices like Google Homes and Amazon Echos deliver wikiHow advice to listeners in need of how-to guidance.

We remark that the quick summaries are indeed used by commercial voice assistants to answer how-to questions. As voice assistants gain popularity, so does the importance of such coherent-paragraph summaries.

A.3 Data Layout

Raw data is available in the supplementary material, in a json format. Each line consists of a single sample, with the following fields

1. Link to the original article
2. Article title
3. Article text
4. Quick summary
5. Split fold (train, dev, or test)

Finally, it also includes *step_headers*: the first line in each step. This is part of the article but might be considered more important, and therefore, it might find further uses by system designers.

A.4 Dataset Statistics

Most dataset statistics appear in Table 1 in the article's main body and are repeated here for completeness. The total number of samples in the WikiSum dataset is $39,775$. On average, each summary consists of 101.2 words, while each article consists of $1,334.2$ words. The average compression ratio is 13.9.

A.5 Evaluation details

We randomly split WikiSum into 35,775 (document, summary) training pairs, as well as 2,000 validation pairs and 2,000 test pairs. The rest of the datasets were downloaded from the HuggingFace dataset repository[12].

All the datasets were evaluated using TextRank[13] and Pegasus-large. The ROUGE scores throughout the paper were calculated using rouge-score[14]. We utilized TextRank to generate three summary sentences. The Pegasus results on Arxiv, Pubmed, and Arxiv were taken from the Pegasus paper. The results on WikiSum were computed by using the Github repository of the Pegasus paper[15]. Pegasus was trained on a single NVIDIA V100 Tensor Core

[9] www.scrapy.org
[10] https://pypi.org/project/beautifulsoup4
[11] https://www.wikihow.com/Write-or-Edit-a-Quick-Summary-on-wikiHow
[12] https://huggingface.co/datasets
[13] https://pypi.org/project/summa
[14] https://pypi.org/project/rouge-score/
[15] https://github.com/google-research/pegasus

GPU, using max input and output sequence lengths of 1024 and 256, respectively.

B Example Summaries

In this appendix, we provide an example summary from WikiSum and arXiv, PubMed, and bigPatent. Note that the article can be quite long (for arXiv and PubMed, it is a full academic paper), so it is not presented in this appendix. Instead, we provide a link to the online version of the full article.

B.1 WikiSum

The WikiSum example summary is provided below:

"To ace a test, even if you're not prepared, start by glancing over the test before you get started to get an idea of how long it is so you can manage your time better. Then, read through each question twice and try to answer it. If you can't answer a question, skip it and come back to it later if you can, which will save you from wasting all of your time on one question. If your test is multiple choice and you don't know the answer, eliminate two answers, so you're left with just two options. Then, guess if necessary since you'll have a 50-percent chance of being right."

The article is available at https://www.wikihow.com/Ace-a-Test.

B.2 WikiHow

For the sake of comparison between WikiHow and WikiSum datasets, we provide the WikiHow summary originating from the same raw material (i.e., the same wikihow.com how-to article) as the WikiSum example at Appendix B.1. We remark that the article to be summarized is not exactly the same, as the WikiHow example does not contain the step headers from the article's text. The WikiHow summary is provided below.

"Study well before the test. Get a study friend. Take breaks. Relax. Pay attention in class. Do all available practice questions. Get some sleep the night before. Have proper meals before the test day. Have your test-taking materials assembled and ready. Listen to music you like. Go into the test in a positive manner. Take deep breaths to try to keep calm. Read the questions carefully. Do the easy questions first. Go with your first answer. Use logic if you're stuck on a multiple choice question. Review your answers thoroughly when you are done."

It can easily be seen that the WikiSum summary is a coherent, fluent paragraph, while the WikiHow summary is a set of bullet points. The content of the two summaries are also quite different between the two datasets.

B.3 arXiv

"the effect of a random phase diffuser on fluctuations of laser light (scintillations) is studied. not only spatial but also temporal phase variations introduced by the phase diffuser are analyzed. the explicit dependence of the scintillation index on finite - time phase variations is obtained for long propagation paths. it is shown that for large amplitudes of phase fluctuations , a finite - time effect decreases the ability of phase diffuser to suppress the scintillations."

The article is available at https://arxiv.org/pdf/0903.5449.pdf.

B.4 PubMed

"tardive dystonia (td) is a serious side effect of antipsychotic medications, more with typical antipsychotics, that is potentially irreversible in affected patients. studies show that newer atypical antipsychotics have a lower risk of td. as a result, many clinicians may have developed a false sense of security when prescribing these medications. we report a case of 20-year - old male with hyperthymic temperament and borderline intellectual functioning, who developed severe td after low dose short duration exposure to atypical antipsychotic risperidone and then olanzapine. the goal of this paper is to alert the reader to be judicious and cautious before using casual low dose second generation antipsychotics in patient with no core psychotic features, hyperthymic temperament, or borderline intellectual functioning suggestive of organic brain damage, who are more prone to develop adverse effects such as td and monitor the onset of td in patients taking atypical antipsychotics."

The article is available at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5330001/.

B.5 BigPatent

"this invention relates to novel calcium phosphate - coated implantable medical devices and processes of making same. the calcium - phosphate coatings are designed to minimize the immune response to the implant and can be used to store and release a medicinally active agent in a controlled manner.

such coatings can be applied to any implantable medical devices and are useful for a number of medical procedures including balloon angioplasty in cardiovascular stenting, ureteral stenting and catheterisation. the calcium phosphate coatings can be applied to a substrate as one or more coatings by a sol - gel deposition process, an aerosol - gel deposition process, a biomimetic deposition process, a calcium phosphate cement deposition process, an electro - phoretic deposition process or an electrochemical deposition process. the coating can contain and elude a drug in an engineered manner."

The article is available at https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2007147234.

UMIC: An Unreferenced Metric for Image Captioning via Contrastive Learning

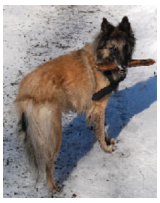
Hwanhee Lee[1], **Seunghyun Yoon**[2], **Franck Dernoncourt**[2]
Trung Bui[2] and **Kyomin Jung**[1]
[1]Dept. of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
[2]Adobe Research, San Jose, CA, USA,
{wanted1007, kjung}@snu.ac.kr
{syoon, franck.dernoncourt, bui}@adobe.com

Abstract

Despite the success of various text generation metrics such as BERTScore, it is still difficult to evaluate the image captions without enough reference captions due to the diversity of the descriptions. In this paper, we introduce a new metric UMIC, an **U**nreferenced **M**etric for **I**mage **C**aptioning which does not require reference captions to evaluate image captions. Based on Vision-and-Language BERT, we train UMIC to discriminate negative captions via contrastive learning. Also, we observe critical problems of the previous benchmark dataset (i.e., human annotations) on image captioning metric, and introduce a new collection of human annotations on the generated captions. We validate UMIC on four datasets, including our new dataset, and show that UMIC has a higher correlation than all previous metrics that require multiple references. We release the benchmark dataset and pre-trained models to compute the UMIC[1].

1 Introduction

Image captioning is a task that aims to generate a description that explains the given image in a natural language. While there have been many advances for caption generation algorithms (Vinyals et al., 2015; Anderson et al., 2018) and target datasets (Fang et al., 2015; Sharma et al., 2018), few studies (Vedantam et al., 2015; Anderson et al., 2016; Cui et al., 2018; Lee et al., 2020) have focused on assessing the quality of the generated captions. Especially, most of the evaluation metrics only use reference captions to evaluate the caption although the main context is an image. However, as shown in Figure 1, since there are many possible reference captions for a single image, a candidate caption can receive completely different scores depending on the type of reference (Yi et al., 2020). Because of this diverse nature of image captions, reference-based metrics usually use multiple references which are difficult to obtain. To overcome this limitation, we propose UMIC, an **U**nreference **M**etric for **I**mage **C**aptioning, which is not dependent on the reference captions and use an image-caption pair to evaluate a caption. We develop UMIC upon UNITER (Chen et al., 2020) which is a state-of-the-arts pre-trained representation for vision-and-language tasks. Since UNITER is pre-trained to predict the alignment for large amounts of image-text pairs, we consider that UNITER can be a strong baseline for developing an unreferenced metric. We fine-tune UNITER via contrastive learning, where the model is trained to compare and discriminate the ground-truth captions and diverse synthetic negative samples. We carefully prepare the negative samples that can represent most of the undesirable cases in captioning, such as *grammatically incorrect*, *irrelevant to the image*, or *relevant but have wrong keyword*.



Ref 1: A dog standing in the snow with a stick in its mouth.
Ref 2: A little dog holding sticks in its mouth.
Candidate: A dog standing on the snow with a dog
CIDEr with Ref 1: 3.166
CIDEr with Ref 2: 0.281
Human Judgments : 1.875 out of 5

Figure 1: An example where the metric score for a given candidate caption varies significantly depending on the reference type.

When evaluating the metric's performance, it is required to compare the correlations between human judgments and the metric's evaluation score for given datasets. We choose three standard benchmark datasets (i.e., Composite (Aditya et al., 2015), Flickr8k (Hodosh et al., 2013), PASCAL-50s (Vedantam et al., 2015)) and further analyze the quality of the dataset. Interestingly, we found that there exist critical issues in the benchmark datasets,

[1]https://github.com/hwanheelee1993/UMIC

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 220–226
August 1–6, 2021. ©2021 Association for Computational Linguistics

such as poor-label or polarized-label. To perform a rigorous evaluation as well as stimulate the research in this area, we collect new 1,000 human judgments for the model-generated caption. Finally, we evaluate our proposed metric on four benchmark datasets, including our new dataset. Experimental results show that our proposed unreferenced metric is highly correlated with human judgments than all of the previous metrics that use reference captions.

2 Related Work

Image Captioning Metrics Following other text generation tasks such as dialogue systems and machine translation, n-gram similarity metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) are widely used to evaluate an image caption. Especially, CIDEr (Vedantam et al., 2015), which weights each n-gram using TF-IDF, is widely used. SPICE (Anderson et al., 2016) is a captioning metric based on scene graph. BERTScore (Zhang et al., 2019), which computes the similarity of the contextualized embeddings, are also used. BERT-TBR (Yi et al., 2020) focuses on the variance in multiple hypothesis and ViLBERTScore (VBTScore) (Lee et al., 2020) utilizes ViLBERT (Lu et al., 2019) to improve BERTScore.

Different from these metrics, VIFIDEL (Madhyastha et al., 2019) computes the word mover distance (Kusner et al., 2015) between the object labels in the image and the candidate captions, and it does not require reference captions. Similar to VIFIDEL, our proposed UMIC does not utilize the reference captions. However, UMIC directly uses image features and evaluates a caption in various perspectives compared to VIFIDEL.

Quality Estimation Quality Estimation (QE) is a task that estimates the quality of the generated text without using the human references and this task is same as developing an unreferenced metric. QE is widely established in machine translation (MT) tasks (Specia et al., 2013; Martins et al., 2017; Specia et al., 2018). Recently, (Levinboim et al., 2021) introduces a large scale human ratings on image-caption pairs for training QE models in image captioning tasks. Our work also trains caption QE model, (i.e. unreferenced captioning metric) but we do not use human ratings to train the metric. Instead, we create diverse synthetic negative samples and train the metric with these samples via ranking loss.

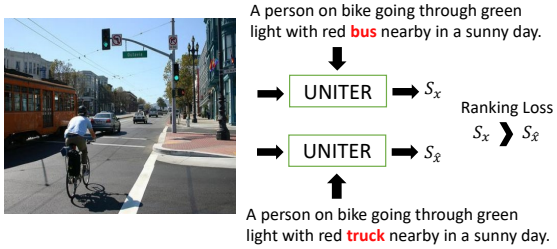


Figure 2: Overall training procedure of UMIC. Given an image I, a positive caption x and a negative caption $\hat{x}$, we compute the score of each image-caption pair S_x and $S_{\hat{x}}$ using UNITER respectively. Then, we fine-tune UNITER using raking loss that S_x is higher than $S_{\hat{x}}$.

3 UMIC

We propose UMIC, an unreferenced metric for image captioning using UNITER. We construct negative captions using the reference captions through the pre-defined rules. Then, we fine-tune UNITER to distinguish the reference captions and these synthetic negative captions to develop UMIC.

3.1 Modeling

Since UNITER is pre-trained to predict the alignment of large amounts of image-text pairs, we use the output of the layer that predicts this alignment as the baseline of UMIC to be fine-tuned. Specifically, we compute the score of a caption $\mathrm{S}(I, X)$ for given image $I = (i_1, ..., i_N)$ and $X = (x_1, ..., x_T)$ as follows.

We first compute the contextual embedding for I and X using UNITER to get the joint representation of image and text as follows.

$$i_{[CLS]}, i_1, ..., i_N, x_1, ..., x_T = \text{UNITER}(I, X), \quad (1)$$

where $i_{[CLS]}$ is a joint representation of the input image and input caption. Then we feed it into a single fully-connected layer to get a score as follows.

$$\mathrm{S}(I, X) = sigmoid(W i_{[CLS]} + b), \quad (2)$$

where W and b are trainable parameters.

3.2 Negative Samples

To model negative captions, we observe the captions' common error types in the model-generated captions. Specifically, we pick 100 bad captions in the order of whose human judgments are low in Composite and Flickr8k, respectively. Then, we categorize the main errors into three types:*relevant but have wrong keywords, totally irrelevant to the image, grammatically incorrect.* To model most

Target Image

Similar Image

Original: a woman hugging a girl who is holding a suitcase

Substitiution: a boy hugging a girl who is holding a suitcase

Random(Hard Negative): a very small cute child by a suitcase

Repetition & Removal: a woman hugging a girl is holding a suitcase suitcase

Figure 3: An example of the generated negative captions for the left image to train UMIC. Hard negative caption is one of the reference captions for the right image which is similar to the left image.

imperfect captions including these frequent type errors, we prepare negative captions as follows.

Substituting Keywords To mimic the captions that are relevant but have wrong keywords, as in the example of Figure 2, we randomly substitute 30% of the words in the reference captions and use them as negative samples like Figure 3. The motivation we choose 30% is that the average length of the generated caption is about 10 words and the number of keywords is usually around three. We only substitute *verb*, *adjective*, and *noun*, which are likely to be keywords since they are usually visual words. Also, we substitute them with the words with the same POS-Tags using the pre-defined dictionaries for the captions in the training set to conserve the sentence structure.

Random Captions We randomly sample captions from other images and use them as negative samples to generate totally irrelevant captions for the given image. Also, similar to the image-text retrieval task, we use hard-negative captions, which are difficult to be discerned, with a probability of 50%. Specifically, we utilize the captions of the images similar to the given images using the pre-trained image retrieval model. We get negative captions that are the captions of the similar image sets computed by image-text retrieval model VSE++ (Faghri et al., 2018) as in (Wang et al., 2020). Then, we sample the captions in the reference captions of the Top-3 similar image sets like the example in Figure 3.

Repetition & Removal We find that some of the captions have repeated words or have incomplete sentences. Hence, we randomly repeat or remove some words in the reference captions with a probability of 30% in the captions to generate these kinds of captions. Specifically, we choose to repeat or remove with a probability of 50% for the sampled word.

Word Order Permutation We further generate negative samples by randomly changing the word order of the reference captions, so that the model sees the overall structure of the sentence, not just the specific visual words.

3.3 Contrastive Learning

Using the negative captions generated by the above rules, we fine-tune UNITER via contrastive loss for positive caption X and negative caption $\hat{X}$ as follows.

$$Loss = max(0, M - (\mathrm{S}(I, X) - \mathrm{S}(I, \hat{X}))), \quad (3)$$

where M is the margin for the ranking loss, which is a hyperparameter. We make each batch composed of one positive caption and four negative captions that are made by each negative sample generation technique.

4 Dataset

We briefly explain the previous benchmark datasets for captioning metrics and analyze the problems for two of these datasets, Flickr8k and Composite. Also, we introduce a new benchmark dataset to alleviate the addressed problems.

4.1 Commonly Used Datasets

Composite consists of 11,985 human judgments for each candidate caption generated from three models and image pair. This dataset's human judgments range from 1 to 5, depending on the relevance between candidate caption and image.

Flickr8k provides three expert annotations for each image and candidate caption on 5,822 images. The score ranges from 1 to 4, depending on how well the caption and image match. All of the captions in this dataset are reference captions or captions from other images.

PASCAL50s contains 1,000 images from UIUC PASCAL Sentence Dataset with 50 reference captions for each image. Different from other datasets, this dataset provides 4,000 caption triplet <*A*, *B*, *C*> composed of 50 reference captions(*A*) and two candidate captions(*B*, *C*) for the given image. There

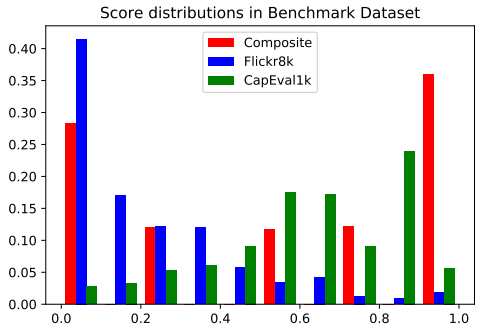


Figure 4: Score distributions of human judgments in Composite, Flickr8k and our proposed CapEval1k dataset. All scores were normalized from 0 to 1.

are human annotated answers to which is more similar to "*A*", "*B*" or "*C*".

4.2 Problems in Flickr8k and Composite

We investigate the human judgments in Flickr8k and Composite, and visualize the distributions of judgment scores for two datasets, Flickr8k and Composite in Figure 4, and find several problems.

For the Flickr8k, most of the scores are less than 0.2 since the candidate captions were sampled by an image retrieval system from a reference caption pool, not model-generated captions. Therefore, most captions are not related to images and differ significantly from the model-generated captions. We argue that this naive configuration is not enough to distinguish the performance of the metric precisely.

For the Composite, most of the scores are placed near 0 or 1. We explain this because only a single annotator annotates each sample's score resulting in biased output. We also manually investigated the captions and found that the captions are coarsely generated. Note that the captions for this dataset were generated by the old model (Karpathy and Fei-Fei, 2015; Aditya et al., 2015). For these reasons, we conclude that additional benchmark dataset is necessary to evaluate the captioning metrics.

4.3 CapEval1k Dataset

To alleviate the addressed issues in Flickr8k and Composite, we introduce a new dataset CapEval1k, which is composed of human judgments for the model-generated captions from four recently proposed models: Att2in (Rennie et al., 2017), Transformer (Vaswani et al., 2017), BUTD (Anderson et al., 2018) and AoANet (Huang et al., 2019). Different from Flickr8k and Composite, we ask each annotator to evaluate the captions by considering three dimensions: *fluency*, *relevance*, *descriptiveness*. We hire 5 workers who are fluent in English for each assignment from Amazon Mechanical Turk and use the average score. We also provide the full instructions and details in Appendix.

Since our CapEval1k dataset is composed of annotations via recently proposed models, the overall scores are relatively higher than other datasets as shown in Figure 4. Compared to other datasets, CapEval1k contains the annotators' comprehensive judgment across multiple dimensions in evaluating the quality of the generated captions, so we can see that the score distribution score is not concentrated in a particular area.

Metric	Flickr8k	Composite	CapEval1k	PASCAL50s
BLEU-1	0.274	0.406	0.233	74.3
BLEU-4	0.286	0.439	0.238	73.4
ROUGE-L	0.300	0.417	0.220	74.9
METEOR	0.403	0.466	0.288	78.5
CIDEr	0.419	0.473	0.307	76.1
SPICE	0.457	0.486	0.279	73.6
BERTScore	0.396	0.456	0.273	79.5
BERT-TBR	0.467	0.439	0.257	**80.1**
VBTScore	**0.525**	**0.514**	**0.352**	79.6
VIFIDEL	0.336	0.191	0.143	70.0
UMIC	**0.468**	**0.561**	**0.328**	**85.1**
UMIC$_C$	0.431	0.554	0.299	84.7

Table 1: Columns 1 to 3 represent Kendall Correlation between human judgments and various metrics on Flickr8k, Composite and CapEval1k. All p-values in the results are < 0.01. The last column shows the accuracy of matches between human judgments in PASCAL50s.

5 Experiments

5.1 Implementation Details

We use the pre-trained UNITER-base with 12 layers in the official code provided by the authors (Chen et al., 2020)[2]. We use the COCO dataset (Fang et al., 2015) to fine-tune UNITER through ranking loss. We use the train and validation split of COCO dataset in (Chen et al., 2020). The number of the training set is 414k, and the validation set is 25k. We set the batch size of 320, learning rate of 2e-6, and fine-tune UNITER for a maximum of 4k steps. We select the model that shows the minimum loss in the validation set. We set margin M as 0.2 in the ranking loss. We repeat training 5 times for each best-performing model.

5.2 Performance Comparison

We compute caption-level Kendall's correlation coefficient with human judgments for the Composite,

[2]https://github.com/ChenRocks/UNITER

Flickr8k, and our proposed CapEval1k. For the PASCAL50s, we compute the number of matches between human judgments for each candidate caption pair. For all of the reference based metrics, we use five reference captions and then get average score among the five references except for BERTScore where we use maximum.

We present the experimental results for all four datasets in Table 1. We show that although UMIC does not utilize any reference captions, UMIC outperforms the baseline metrics except for VBTScore in all of the datasets that depend on multiple references. We also report the strong unreferenced baseline UMIC$_{\text{-C}}$, which is directly using the pre-trained weights from UNITER without contrastive learning. Interestingly, UMIC$_{\text{-C}}$ shows a higher performance than most of the metrics. This high performance shows that pre-trained image-text matching layer of UNITER already has a good representation for evaluating image captions. Especially for Composite, both UMIC and UMIC$_{\text{-C}}$ significantly outperform baseline metrics. We explain this in the polarized distribution of human judgments as we explained in Section 4.2. In other words, the relevance of most image-caption pairs in this dataset is too obvious so that UNITER can easily distinguish them. However, while UMIC shows higher performance on all datasets, UMIC$_{\text{-C}}$ shows relatively low performance on Flickr8k and CapEval1k. And this demonstrates the effectiveness and generalization ability of our contrastive learning objective to develop UMIC.

Also, we can observe that the performance of each metric is relatively low and the rank of each metric changes in our proposed CapEval1k dataset. We explain that this is because the captions in CapEval1k are relatively difficult to be evaluated since the score distribution is not biased as explained in Section 4.3.

5.3 Case Study

We visualize one sample each showing the strengths and weaknesses of UMIC in Figure 5. In the above example, the candidate caption is partially relevant to the image, but the single word "three" in the caption is totally incorrect since there are only "two" giraffes in the image. And this leads to a low human judgment of 0.2. Nevertheless, unlike our UMIC, widely used metrics and UMIC$_{\text{-C}}$ give this caption a high score due to the many words overlaps or missing the keywords. The bottom example shows one of the error cases and the limitations of our proposed method. Since the detection model in UMIC could not recognize the important object like the "baseball bat", UMIC outputs very low score.

References
- two giraffe standing next to each other in a field.
- two giraffes are climbing a hill with mountains in the background.

Candidate
- **three** giraffes standing in a field of grass

BLEU1: 0.324	**ROUGE-L**: 0.320	**METEOR**: 0.173	**CIDER**: 0.866
SPICE: 0.289	**UMIC**: 0.352	**UMIC**$_{/-C}$: 0.770	**Human**: 0.200

References
- a person breadking a bottle with a baseball bat
- a boy in yellow shirt swinging a baseball bat

Candidate
- a man swinging a **baseball bat** at a ball

BLEU1: 0.360	**ROUGE-L**: 0.354	**METEOR**: 0.176	**CIDER**: 1.205
SPICE: 0.192	**UMIC**: 0.094	**UMIC**$_{/-C}$: 0.062	**Human**: 0.450

Figure 5: Case study for the various metrics on candidate captions in CapEval1k Dataset. Human judgments are normarlized from 0 to 1.

6 Conclusion

In this paper, we propose UMIC, an unreferened metric that does not require any reference captions for image captioning task through contrastive learning in UNITER. Also, we propose a new benchmark dataset for image captioning that relieve the issues in previous datasets. Experimental results on four benchmark datasets, including our new dataset, show that UMIC outperforms previous metrics.

Acknowledgements

We thank anonymous reviewers for their constructive and insightful comments. K. Jung is with ASRI, Seoul National University, Korea. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2021R1A2C2008855). This work was partially funded by gifts from Adobe Research.

Ethical Considerations

We compensate the annotators with competitive pay, which is above hourly USD $10 for collecting human annotated judgments for the model generated captions. Specifically, we pay $0.2 for each task that is composed of evaluating four candidate captions for a single image, where each task can be usually done in a minute. And we use public datasets to train the models.

References

Somak Aditya, Yezhou Yang, Chitta Baral, Cornelia Fermuller, and Yiannis Aloimonos. 2015. From images to sentences through scene description graphs using commonsense reasoning and knowledge. *arXiv preprint arXiv:1511.03292*.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. 2018. Learning to evaluate image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5804–5812.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. Vse++: Improving visual-semantic embeddings with hard negatives.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2020. Vilbertscore: Evaluating image caption using vision-and-language bert. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 34–39.

Tomer Levinboim, Ashish V. Thapliyal, Piyush Sharma, and Radu Soricut. 2021. Quality estimation for image captions based on large-scale human evaluations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3157–3166, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Pranava Swaroop Madhyastha, Josiah Wang, and Lucia Specia. 2019. Vifidel: Evaluating the visual fidelity of image descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6539–6550.

André FT Martins, Marcin Junczys-Dowmunt, Fabio N Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André FT Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709.

Lucia Specia, Kashif Shah, José GC De Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Jiuniu Wang, Wenjia Xu, Qingzhong Wang, and Antoni B Chan. 2020. Compare and reweight: Distinctive image captioning using similar images sets. In *ECCV*.

Yanzhi Yi, Hangyu Deng, and Jinglu Hu. 2020. Improving image captioning evaluation by considering inter references variance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 985–994.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Anchor-based Bilingual Word Embeddings for Low-Resource Languages

Tobias Eder[1], **Viktor Hangya**[2] and **Alexander Fraser**[2]
[1]Research Group Social Computing
Technical University of Munich
[2]Center for Information and Language Processing
LMU Munich
tobias.eder@in.tum.de,
{hangyav, fraser}@cis.lmu.de

Abstract

Good quality monolingual word embeddings (MWEs) can be built for languages which have large amounts of unlabeled text. MWEs can be aligned to bilingual spaces using only a few thousand word translation pairs. For low resource languages training MWEs monolingually results in MWEs of poor quality, and thus poor bilingual word embeddings (BWEs) as well. This paper proposes a new approach for building BWEs in which the vector space of the high resource source language is used as a starting point for training an embedding space for the low resource target language. By using the source vectors as anchors the vector spaces are automatically aligned during training. We experiment on English-German, English-Hiligaynon and English-Macedonian. We show that our approach results not only in improved BWEs and bilingual lexicon induction performance, but also in improved target language MWE quality as measured using monolingual word similarity.

1 Introduction

Bilingual Word Embeddings are useful for cross-lingual tasks such as cross-lingual transfer learning or machine translation. Mapping based BWE approaches rely only on a cheap bilingual signal, in the form of a seed lexicon, and monolingual data to train monolingual word embeddings (MWEs) for each language, which makes them easily applicable in low-resource scenarios (Mikolov et al., 2013b; Xing et al., 2015; Artetxe et al., 2016). It was shown that BWEs can be built using a small seed lexicon (Artetxe et al., 2017) or without any word pairs (Lample et al., 2018a; Artetxe et al., 2018) relying on the assumption of isomorphic MWE spaces. Recent approaches showed that BWEs can be built without the mapping step. Lample et al. (2018b) built FASTTEXT embeddings (Bojanowski et al., 2017) on the concatenated source and target language corpora exploiting the shared character n-grams in them. Similarly, the shared source and target language subword tokens are used as a cheap cross-lingual signal in Devlin et al. (2019); Conneau and Lample (2019). Furthermore, the advantages of mapping and jointly training the MWEs and BWEs were combined in Wang et al. (2020) for even better BWEs.

While these approaches already try to minimize the amount of bilingual signal needed for cross-lingual applications, they still require a larger amount of monolingual data to train semantically rich word embeddings (Adams et al., 2017). This becomes a problem when one of the two languages does not have sufficient monolingual data available (Artetxe et al., 2020). In this case, training a good embedding space can be infeasible which means mapping based approaches are not able to build useful BWEs (Michel et al., 2020).

In this paper we introduce a new approach to building BWEs when one of the languages only has limited available monolingual data. Instead of using mapping or joint approaches, this paper takes the middle ground by making use of the MWEs of a resource rich language and training the low resource language embeddings on top of it. For this, a bilingual seed lexicon is used to initialize the representation of target language words by taking the pre-trained vectors of their source pairs prior to target side training, which acts as an informed starting point to shape the vector space during the process. We randomly initialize the representations of all non-lexicon target words and run Continuous Bag-of-Words (CBOW) and skip-gram (SG) training procedures to generate target embeddings with both WORD2VEC (Mikolov et al., 2013a) and FASTTEXT (Bojanowski et al., 2017). Our approach ensures that the source language MWE space is intact, so that the data deficit on the target side does not result in lowered source

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 227–232
August 1–6, 2021. ©2021 Association for Computational Linguistics

embedding quality. The improved monolingual word embeddings for the target language outperform embeddings trained solely on monolingual data for semantic tasks such as word-similarity prediction. We study low-resource settings for English-German and English-Hiligaynon, where previous approaches have failed (Michel et al., 2020), as well as English-Macedonian.

2 Method

Previous mapping approaches rely on the alignment of two pre-trained monolingual word embedding spaces. In case one of the two languages has significantly fewer resources available, this will strongly affect the resulting mapping negatively. This is also an issue for joint approaches because the shared token representations are biased towards the language with more training samples. Our approach instead leverages the high resource language to improve performance on the low-resource language.

We pre-train MWEs for the source language and use the source MWEs to initialize the space of the low resource target language. Using a set of initial seed pairs, the representation of a seed word in the target space is replaced with the representation of its translation (anchor points). Then, training is performed on the initialized space using only monolingual data from the low resource language by only updating the representation of non-seed words which are initialized randomly. Through this method a BWE representation is directly induced from the anchor points of the fixed vectors.

In some cases there are multiple valid translations for a single target language word. We experiment with either initializing with the average over these possible translations or randomly selecting only one of them. The averaging helps by finding a common anchor for the different semantic nuances the token might represent in different target language contexts. Additionally, we experiment with enabling or disabling the updates of anchor vectors during training. We implemented the anchor point based initialization in both WORD2VEC and FASTTEXT with only complete token representations serving as potential anchors. In the case of FASTTEXT these initializations have no influence on the subword (character n-gram) embeddings which are still initialized randomly, which makes intuitive sense in the common case of morphologically different language pairs. Training is performed using standard hyperparameters included in the GENSIM WORD2VEC and FASTTEXT packages (Řehůřek and Sojka, 2010). Unless stated otherwise, vectors are of dimensionality 300 with a context window of 5 words used during training. All models are trained for 5 epochs without further hyperparameter tuning utilizing a single desktop machine on a Intel Core i7-7700K CPU with 4.20Ghz, a NVIDIA GeForce GTX 1080 Ti graphics card and 32 GB of DDR4 SDRAM. The parameters of each trained model are equal to the standard implementation of the packages as listed above. Training time is largely dependent on input size, but corresponds to a few seconds up to roughly 5 minutes in the low resource setting.

2.1 Experimental Setup

First we conduct experiments on the German and English language pair, since large available corpora made it easier to test different sized dictionaries and corpora during training. The basic setup trains a MWE on the source language (English) up front. For this training the WMT 2019 News Crawl corpus in English, including approximately 532 million tokens, was chosen (Barrault et al., 2019). Similarly for the target language, we used the German WMT 2019 News Crawl from which we uniformly sample to obtain training sets of different sizes. All dataset are tokenized and lowercased before training.

To evaluate, we translate German words to English. We use the MUSE German-English dictionary (Lample et al., 2018a). There are 102K translation pairs with a total of roughly 68K unique German words. For each German word there might be multiple valid English translations, which are listed in the dictionary. For the initialization we select either randomly one translation option or the averaged word representations of all available translations, as discussed in section 2. However, many German words have only one valid translation. We used the MUSE test set containing roughly 3000 translation pairs in the frequency range 5000-6500, leaving 99K pairs as potential candidates for the initialization. In our experiments we mostly consider setups with much smaller training lexicon sizes, by taking the top-n most frequent source words and their translations from the lexicon.

In addition to the German experiments we test our system on two lower resource languages: Macedonian and Hiligaynon. For Macedonian we use data in the form of a Wikipedia dump, as well

as the MUSE dictionary for the language pair Macedonian-English for our test setup.[1] For Hiligaynon we use a corpus containing roughly 350K tokens as well as a corresponding dictionary containing 1100 translated terms between English and Hiligaynon and an additional test set of 200 terms released by Michel et al. (2020).

After training, bilingual lexicon induction (BLI) is done by taking the top n closest vectors measured by cross-domain similarity local scaling (CSLS) distance. For better comparability we use the evaluation method provided by MUSE (Lample et al., 2018a) for both the comparison baseline as well as our system. For Hiligaynon we use cosine to compare directly with (Michel et al., 2020).

3 Results

The following section evaluates different models quantitatively using $acc@5$ and $acc@1$ as a metric. The baseline runs MUSE tool in supervised mode using iterative procrustes refinement to obtain the mapping using default parameters as reported in Lample et al. (2018a). For the English embedding the full corpus size was used, while in the case of the (low-resource) languages the corpora sizes were varied to observe changes in performance.

3.1 Bilingual experiments

BLI was performed using the method from section 2. Since Word2Vec SG and FastText embeddings performed much worse with the anchored training, all following numbers report Word2Vec CBOW embeddings.

Table 1 shows the comparison between four different possible setups for the proposed method as explained in Section 2: Either fixing anchor-vectors or allowing them to train or initializing with single word vectors or averaged ones. The overall best performing model utilizes averaged and non-fixed anchor vectors. Table 1 also shows the baselines at varying corpora sizes. Overall the anchor method performs much better than the baseline at lower corpora sizes and stays competitive as corpus size increases. Results are similarly consistent when looking at either $acc@5$ or $acc@1$.

One important parameter for the proposed method is how it scales with the available amount of anchor-vectors used for training. In a range of experiments, different initialization sizes were compared. Figure 1 shows the result for varying the number of anchor vectors. The general trend is that the more anchor vectors, the better the performance, which slowly caps off at the higher end, as more vectors of lower frequency words start introducing noise. The same development is not true for the baseline, which does not benefit equally from increasing the potential seed lexicon vocabulary and even starts losing performance at larger dictionary sizes.

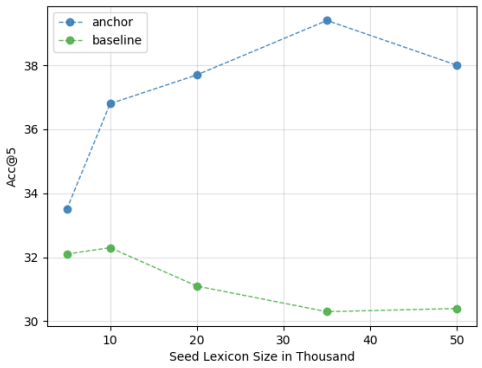


Figure 1: Anchor method for English-German with fixed vectors and baseline with varying training-dictionary sizes at corpus size 20 million

This difference is likely rooted in the inclusion of less frequent word pairs in the larger dictionaries. These words have worse quality representations which introduces noise in the mapping process, thus restricting the precision of their orthogonal alignment as described by (Søgaard et al., 2018). In contrast, our method initializes all target language word embeddings given their pairs, i.e., perfectly aligning all words in the training dictionary, which serve as high quality anchor points for the remaining words.

3.2 Macedonian and Hiligaynon

Another set of experiments was done on the language pair English-Macedonian, a language that already offers less resources than German and is also more dissimilar from English.

Results for experiments comparing between the MUSE baseline and the anchor method are shown in Table 2. The best performing model again combines averaged initialization with trainable anchors.

Compared to the previous experiments with German, results for Macedonian are similar, while the baseline model is overall weaker than before, suggesting the anchor method benefits more strongly

[1] https://dumps.wikimedia.org/mkwiki/ (downloaded on 01/31/21)

Model // Corpus Size	100K	300K	500K	1M	2M	5M	10M	20M	50M
Baseline Word2Vec (CBOW)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.3 (0.0)	0.6 (0.2)	6.1 (1.4)	15.9 (4.2)	26.7 (12.8)	40.2 (21.3)
Baseline FastText (SG)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.6 (0.2)	1.9 (0.5)	8.9 (3.9)	19.7 (8.6)	32.1 (18.1)	45.1 (28.9)
Fixed not Averaged (CBOW)	1.3 (0.9)	0.8 (0.3)	0.8 (0.3)	1.5 (0.5)	4.1 (1.7)	13.6 (5.6)	23.9 (12.2)	26.3 (13.4)	35.7 (21.0)
Fixed and Averaged (CBOW)	1.3 (0.4)	1.2 (0.1)	1.8 (0.2)	1.8 (0.6)	4.3 (1.9)	13.1 (5.5)	23.3 (11.9)	33.2 (18.4)	44.1 (26.3)
Trained not Averaged (CBOW)	1.3 (0.9)	0.8 (0.4)	1.4 (0.8)	3.3 (0.5)	7.3 (1.7)	18.0 (5.6)	27.0 (12.2)	35.8 (21.0)	45.2 (28.4)
Trained and Averaged (CBOW)	1.7 (0.4)	1.5 (0.5)	2.9 (1.4)	4.5 (1.3)	10.7 (4.8)	22.3 (11.9)	32.2 (18.3)	40.5 (25.0)	48.5 (31.5)

Table 1: Evaluation of models at varying data-sizes for English-German. Baselines (MUSE) and proposed methods, reporting acc@5 (acc@1).

Model // Corpus Size	1M	2M	5M	10M	20M	37M
Baseline Word2Vec (CBOW)	0.0 (0.0)	0.7 (0.0)	6.0 (1.9)	18.1 (7.9)	28.0 (15.1)	37.0 (20.4)
Trained and Averaged (CBOW)	1.6 (0.4)	4.7 (1.6)	13.1 (5.6)	26.3 (13.4)	35.5 (18.4)	44.7 (24.2)

Table 2: Anchor method vs. baseline (MUSE) at varying data sizes reporting acc@5 (acc@1) for English-Macedonian.

Algorithm	Acc@5
Baseline Michel et al. 50D	0.5
Baseline Michel et al. 300D	0.0
Anchor fixed not averaged CBOW 300D	2.6
Anchor trained not averaged CBOW 300D	4.3

Table 3: BLI on Hiligaynon-English.

from a high-resource embedding, even when language pairs become more dissimilar.

For English-Hiligaynon previous approaches failed due to limitations of the available monolingual training data. Table 3 shows the performance for translating Hiligaynon words to English. The evaluation was done using cosine-distance for better comparability between the Michel et al. (2020) paper and our results. Since there were only single translations of words in the provided dictionary, the method of averaging vectors for initialization was not used. Similarly, during evaluation only one valid term per word was possible. While Michel et al. (2020) reported 0.5% for 50 dimensional vectors, in our baseline the 50 dimensional vectors achieved a constant 0 (not shown). The numbers are comparable to the low frequency experiments between German and English as seen in Table 1.

3.3 Monolingual experiments

In addition to better BWEs, our approach also improves the low-resource embedding for purely monolingual tasks. To confirm this, the anchor-vector trained embeddings for German were evaluated on monolingual word similarity and compared to the results achieved by regular training of the embedding space. We evaluate on multiple datasets: GUR350 and GUR65 (Gurevych, 2005), SEMEVAL17 (Cer et al., 2017), SIMLEX-999 (Leviant and Reichart, 2015), WS-353 (Agirre et al., 2009) and ZG22 (Zesch and Gurevych, 2006), and report the averaged Spearman's rho correlation between cosine similarity of vector pairs and human annotations. Similar monolingual datasets are not available for Macedonian and Hiligaynon. In Figure 2 the effect of employing the anchor method on monolingual word similarity performance is compared against Word2Vec CBOW trained without anchor initialization. The improvements across different training corpora sizes are in favor of the proposed method, suggesting that it can be employed to improve performance on monolingual tasks. Overall this serves to demonstrate the advantage of the anchor-method on small datasets and allows to learn better monolingual representation from the same amount of data by utilizing the information from a pretrained embedding for a completely different language with more readily available training data. The thus learned representation can not only serve as an already aligned space for translation tasks as shown above, but is also the better performing representation of the monolingual space.

4 Conclusion

We proposed a novel approach to build BWEs to improve performance on language pairs with limited monolingual data on the target side. By utilizing pre-trained MWEs of resource rich languages and a seed lexicon to fix anchor points before training, a structurally similar embedding space can be learned for the low resource language which is aligned with the source representations. We evaluated our approach on the BDI task using English-German to test varying training parameters and corpora sizes, on English-

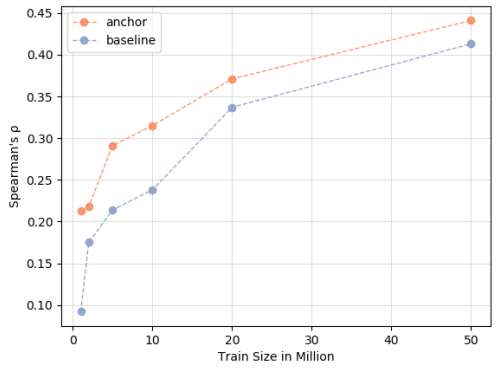


Figure 2: Spearman's rho correlation on monolingual word similarity across corpora sizes for German.

Macedonian and the extremely low resource language pair English-Hiligaynon on which previous approaches failed. We showed that the performance of existing mapping approaches degrades drastically with lower monolingual data sizes, even when there are large seed lexicons available. In contrast, our proposed system outperformed previous mapping based approaches on these setups including English-Hiligaynon. On top of improved BWEs, we showed improved MWE quality as well for the target language by outperforming standard MWEs on the monolingual word similarity task showing that it is beneficial for monolingual tasks as well. We implemented our approach for both Word2Vec and FastText which we publicly release to promote reproducibility and further research.[2]

Ethical Considerations

The proposed system acts as a tool to specifically help in the low resource setting that predominantly affects less researched languages. Even though part of the experiments were done on the higher resource language pair English-German, the results were further confirmed for other pairs of languages.

As a word embedding based system, the resulting mappings and embedding spaces are highly affected by the kind of monolingual content that goes into their training, which is why we made sure to train the embeddings on texts that should adhere to a higher standard, such as verified news media and online articles, instead of a general web crawl. Additionally the seed lexicons used come from verified sources, such as the popular MUSE lexicons in the case of English-German and English-Macedonian as well as from translations by a native speaker of Hiligaynon in the case of English-Hiligaynon.

We hope that in general the proposed methods can help alleviating some of the resource problems less researched languages are facing and thus to close the gap for language technology working with and on these languages.

As part of the ethical responsibility to ensure reproducibility and responsibility in terms of computational resources, we reported results with a set of standard hyperparameters instead of searching for the most optimal setting for our proposed method. Our models are as lightweight as regular training methods for word embeddings and are therefore not very demanding in terms of computation. This is especially true in the low-resource setting, where training time is reduced to just a fraction compared to the bigger corpora.

[2]http://cistern.cis.lmu.de/anchor-embeddings

Acknowledgments

The work was supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 640550) and by German Research Foundation (DFG; grant FR 2829/4-1).

References

Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. 2017. Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 937–947.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT 2009*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing"*, pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised

cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 1–14.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, pages 7057—-7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Iryna Gurevych. 2005. Using the structure of a conceptual network in computing semantic relatedness. In *International Conference on Natural Language Processing*, pages 767–778.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018a. Word Translation Without Parallel Data. In *International Conference on Learning Representations*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv preprint arXiv:1508.00106*.

Leah Michel, Viktor Hangya, and Alexander Fraser. 2020. Exploring bilingual word embeddings for Hiligaynon, a low-resource language. In *LREC 2020, Twelfth International Conference on Language Resources and Evaluation*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. 2020. Cross-lingual Alignment vs Joint Training: A Comparative Study and A Simple Unified Framework. In *Proceedings of the 8th International Conference on Learning Representations*.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011.

Torsten Zesch and Iryna Gurevych. 2006. Automatically creating datasets for measures of semantic relatedness. In *Proceedings of the Workshop on Linguistic Distances*, pages 16–24.

Multilingual Agreement for Multilingual Neural Machine Translation

Jian Yang[1] [*], **Yuwei Yin**[1] [*], **Shuming Ma**[2], **Haoyang Huang**[2],
Dongdong Zhang[2], **Zhoujun Li**[1] [†], **Furu Wei**[2]
[1]State Key Lab of Software Development Environment, Beihang University
[2]Microsoft Research Asia
{jiaya, yuweiyin, lizj}@buaa.edu.cn; {shumma, haohua, dozhang, fuwei}@microsoft.com

Abstract

Although multilingual neural machine translation (MNMT) enables multiple language translations, the training process is based on independent multilingual objectives. Most multilingual models can not explicitly exploit different language pairs to assist each other, ignoring the relationships among them. In this work, we propose a novel agreement-based method to encourage multilingual agreement among different translation directions, which minimizes the differences among them. We combine the multilingual training objectives with the agreement term by randomly substituting some fragments of the source language with their counterpart translations of auxiliary languages. To examine the effectiveness of our method, we conduct experiments on the multilingual translation task of 10 language pairs. Experimental results show that our method achieves significant improvements over the previous multilingual baselines.

1 Introduction

Multilingual neural machine translation (MNMT) has experienced rapid growth in recent years (Johnson et al., 2017; Zhang et al., 2020; Aharoni et al., 2019; Wang et al., 2019). It is not only capable of translating among multiple language pairs by encouraging the crosslingual knowledge transfer to improve low-resource translation performance (Firat et al., 2016b; Zoph et al., 2016; Sen et al., 2019; Qin et al., 2020; Hedderich et al., 2020; Raffel et al., 2020), but also can handle multiple language pairs in a single model, reducing model parameters and training costs (Firat et al., 2016a; Blackwood et al., 2018; Wang et al., 2020; Sun et al., 2020).

Previous works in MNMT simply optimize independent translation objectives and do not use arbitrary auxiliary languages to encourage the agreement across different translation directions. As shown in Figure 1, the multilingual baseline is separately trained on French-English and German-English directions and cannot explicitly promote each other. The German-English translation only implicitly helps the French-English translation since both translation directions share the same encoder. There still exists a gap between German-English and French-English translation directions. As a result, minimizing the difference across different translation directions by an explicit paradigm requires further exploration.

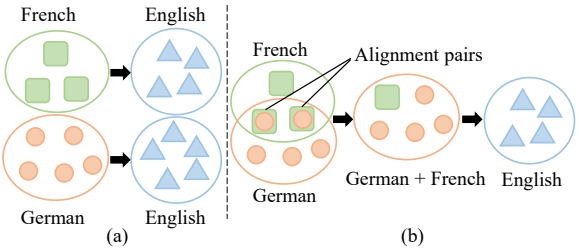


Figure 1: Comparison between (a) the multilingual translation and (b) our agreement-based method.

In this paper, we propose a novel agreement-based method, which explicitly models the shared semantic space for multiple languages and encourages the agreement across them. Our training procedure extends the multilingual translation with the agreement term, which encourages the model to produce the source sentence with multiple languages into the target sentence. As Figure 1 shows, we randomly substitute some source phrases with their counterparts of other languages to create code-switched sentences using word alignment. Our model is jointly trained with the multilingual translation and agreement objectives, where the code-switched sentences are translated into the target sentences. The key idea is to encourage the agreement among different translation directions simul-

[*]Contribution during internship at Microsoft Research Asia.
[†]Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 233–239
August 1–6, 2021. ©2021 Association for Computational Linguistics

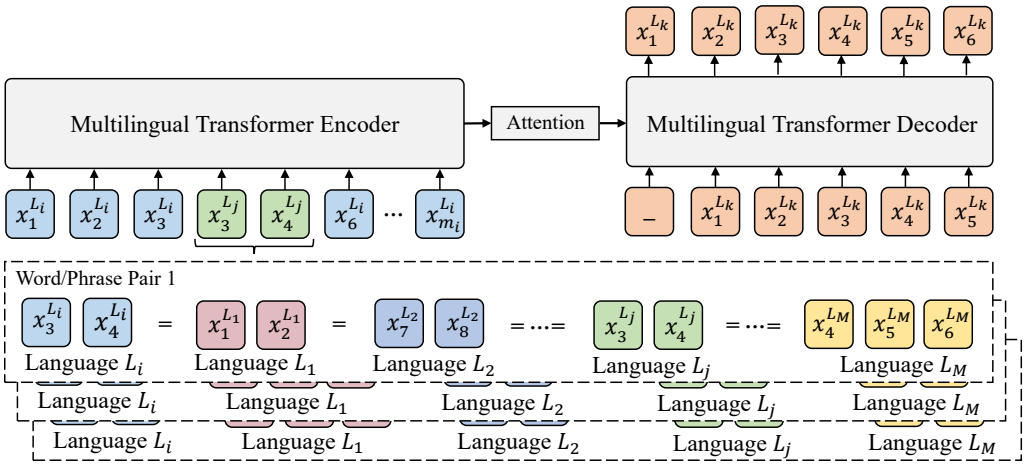


Figure 2: Overview of our method. $x_{m_i}^{L_i}$ denotes the m_i-th token in the sentence of language L_i. We randomly substitute source phrases of language $L_{src} = L_i$ with the translations of other languages $L_{aux} \in L_{all}$ to create code-switched sentences. Different words/phrases with the same meanings may contain different numbers of tokens. Then the code-switched source sentences are translated to the target language $L_{tgt} = L_k$ by the multilingual model. This process greatly encourages multilingual agreement across different translation directions.

taneously by leveraging alignment information of the bilingual source sentence pairs.

Experimental results on the multilingual translation task of WMT demonstrate that our method outperforms the multilingual baseline by a large margin. To better explain the BLEU improvements, we visualize the sentence-level crosslingual representations and the attention weights across different languages, which shows that our method effectively encourages the agreement between languages.

2 Our Approach

2.1 Multilingual Machine Translation

Our multilingual model is based on the single Transformer model (Vaswani et al., 2017) and shares all embedding matrices by a common vocabulary of all languages. Given M languages $L_{all} = \{L_1, \ldots, L_M\}$, the multilingual model appends special symbols to the source text to indicate the translation direction from the source language L_{src} to the target language L_{tgt}.

2.2 Agreement-based Training

Multilingual models can translate multiple source-side languages into target-side languages. Given N bilingual corpora $D_B = \{D_{B_1}, \ldots, D_{B_N}\}$, the multilingual model with parameters θ is jointly trained over N language directions to optimize the combined objective as below:

$$\mathcal{L}_{MT} = \sum_{n=1}^{N} \mathbb{E}_{x,y \in D_{B_n}} [-\log P_\theta(y|x)] \quad (1)$$

where x, y denote the sentence pair in the bilingual corpus D_{B_n}. $\mathcal{L}_{MT}$ is the combined translation objective of the multilingual model.

The agreement objective over the code-switched corpora D_C is calculated by:

$$\mathcal{L}_{AT} = \mathbb{E}_{x^{L_{src}/L_{aux}}, y \in D_C} [-\log P_\theta(y|x^{L_{src}/L_{aux}})] \quad (2)$$

where $x^{L_{src}/L_{aux}}$ is the code-switched sentence in which some phrases are substituted by their counterpart phrases in other languages and y is the target sentence. L_{aux} is the auxiliary language.

We combine the bilingual corpora D_B and code-switched corpora D_C to train our agreement-based model, which minimizes the gaps among different translation directions using word alignment:

$$\mathcal{L}_{ALL} = \mathcal{L}_{MT} + \mathcal{L}_{AT} \quad (3)$$

where $\mathcal{L}_{ALL}$ is the combined objective.

2.3 Constructing Training Samples

We use L_{src} as the source language, L_{tgt} as target language, and L_{aux} as auxiliary languages to construct training samples. As shown in Figure 2, $x^{L_{src}} = (x_1^{L_{src}}, \ldots, x_m^{L_{src}})$ is the source sentence with m tokens and $x^{L_{aux}} = (x_1^{L_{aux}}, \ldots, x_n^{L_{aux}})$ is the auxiliary sentence with n tokens. $x_{u:v}^{L_{src}}$ denotes the sentence fragment of $x^{L_{src}}$ from the u-th to v-th token and $x_{s:t}^{L_{aux}}$ denotes the fragment of $x^{L_{aux}}$ from the s-th to t-th token, where $x_{s:t}^{L_{aux}}$ of language L_{aux} is the translation of the $x_{u:v}^{L_{src}}$ of language L_{src}. Formally, the code-switched sequence

En → X	Fr	Cs	De	Fi	Lv	Et	Ro	Hi	Tr	Gu	Avg
Bilingual NMT	36.3	22.3	40.2	15.2	16.5	15.0	23.0	12.2	13.3	7.9	20.2
One-to-Many	34.2	20.9	40.0	15.0	18.1	20.9	26.0	14.5	17.3	13.2	22.0
One-to-Many + Pseudo	35.5	21.7	42.0	16.4	19.3	22.0	26.6	16.2	17.9	17.8	23.5
One-to-Many + AT (our method)	35.7	22.0	42.1	16.6	20.1	22.2	26.9	16.6	18.2	17.9	**23.9**

Table 1: En→X test results for bilingual and multilingual models of 10 language pairs on the WMT benchmark.

X → En	Fr	Cs	De	Fi	Lv	Et	Ro	Hi	Tr	Gu	Avg
Bilingual NMT	36.2	28.5	40.2	19.2	17.5	19.7	29.8	14.1	15.1	9.3	23.0
Many-to-One	34.8	29.0	40.1	21.2	20.4	26.2	34.8	22.8	23.8	19.2	27.2
Many-to-One + Pseudo	35.4	30.1	42.1	22.0	21.2	29.0	35.8	27.3	26.0	22.6	29.1
Many-to-One + AT (our method)	35.7	30.2	42.6	22.3	21.8	29.5	36.4	27.6	26.7	22.8	**29.6**

Table 2: X→En test results for bilingual and multilingual models of 10 language pairs on the WMT benchmark.

$x^{L_{src}/L_{aux}}$ is described as:

$$x^{L_{src}/L_{aux}} = (x_1^{L_{src}}, \ldots, x_{s:t}^{L_{aux}}, \ldots, x_m^{L_{src}}) \quad (4)$$

where most words in the code-switched sentence $x^{L_{src}/L_{aux}}$ are derived from $x^{L_{src}}$, while some source phrases $x_{u:v}^{L_{src}}$ are substituted by their counterpart phrases $x_{s:t}^{L_{aux}}$.

Given the parallel sentences among M different languages, we can construct code-switched source sentence $x^{L_{src}/L_{aux}}$ with different auxiliary languages. Therefore, the code-switched corpora D_C can be constructed in a similar way for other languages to encourage the agreement across different translation directions to help each other.

3 Experiment Setup

3.1 Multilingual Data

We use the same training, valid, and test sets as the previous work (Wang et al., 2020) to evaluate multilingual models by parallel data from multiple WMT datasets with various languages, including English (En), French (Fr), Czech (Cs), German (De), Finnish (Fi), Latvian (Lv), Estonian (Et), Romanian (Ro), Hindi (Hi), Turkish (Tr), and Gujarati (Gu). For each language, we concatenate the WMT data of the latest available year and get at most 10M sentences by randomly sampling. Detailed statistics of datasets are listed in Table 3. All sentences in our experiments are tokenized by SentencePiece[1] (Kudo and Richardson, 2018).

	Train Size	Valid	Test
En-Fr	10.00M	newstest13	newstest15
En-Cs	10.00M	newstest16	newstest18
En-De	4.60M	newstest16	newstest18
En-Fi	4.80M	newstest16	newstest18
En-Lv	1.40M	newsdev17	newstest17
En-Et	0.70M	newsdev18	newstest18
En-Ro	0.50M	newsdev16	newstest16
En-Hi	0.26M	newsdev14	newstest14
En-Tr	0.18M	newstest16	newstest18
En-Gu	0.08M	newsdev19	newstest19

Table 3: The statistics of the training, valid, and test sets on WMT datasets of 10 language pairs.

3.2 Baselines and Evaluation

We compare our method against the following baselines. **Bilingual baseline** is trained on each language pair separately. **One-to-Many** and **Many-to-One** are trained on the En→X and X→En directions respectively. We collect all English sentences (33M) of the bilingual corpora described above and translate them into other languages sentences. We extract alignment pairs (Dyer et al., 2013) across different languages for our method. **One-to-Many + Pseudo** and **Many-to-One + Pseudo** are trained on multilingual data combined with the pseudo data. We average the last 5 checkpoints and employ the beam search strategy with a beam size of 5 for evaluation. The evaluation metric is case-sensitive detokenized sacreBLEU[2] (Post, 2018).

[1] https://github.com/google/sentencepiece

[2] BLEU+case.mixed+lang.{src}-{tgt}+numrefs.1+smooth.exp+tok.13a+version.1.4.14

3.3 Training Details

We adopt the `Transformer_big` architecture as the backbone model for all our experiments, which has 6 layers with an embedding size of 1024, a dropout of 0.1, the feed-forward network size of 4096, and 16 attention heads. We train multilingual models with Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.98$). The learning rate is set as 5e-4 with a warm-up step of 4,000. The models are trained with the label smoothing cross-entropy with a smoothing ratio of 0.1. The batch size is 5,120 tokens and the parameters are updated every 16 iterations to simulate a 128-GPU environment.

4 Results

The results of our model are separately listed in Table 1 and Table 2. Table 1 shows that **One-to-Many** outperforms **bilingual NMT** by +1.8 BLEU points on average. Our method further improves over both **One-to-Many** and **One-to-Many + Pseudo** consistently. Using pseudo and code-switched data brings more improvements to the low-resource languages (Et, Ro, Hi, Tr, and Gu) than high-resource languages (Fr, Cs, De, Fi, and Lv). These results suggest that our model encourages the agreement between different translation directions.

Table 2 reports the results on the X→En test sets. **Many-to-One** outperforms the **bilingual** NMT by +4.2 BLEU points on average. We combine the parallel data with the pseudo data, leading to an improvement of +1.9 BLEU points over **Many-to-One**. Our method further outperforms **Many-to-One + Pseudo** by a large gain of +0.5 BLEU points on average, showing the effectiveness of our agreement-based method and the significance of multilingual agreement.

5 Analysis

Attention Visualization The representations of attention in Figures 3 and 4 are averaged over all 16 heads of the last layer. Figure 3 shows the self-attention weights of a code-switched English sentence, where the source phrase "coordination between law enforcement" is substituted by the German phrase "Koordinierung zwischen Strafverfolgung sbehörden". Similar to the common attention pattern, our model can learn better crosslingual representations in this code-switching case. Figure 4 shows that the cross-attention weights between the input code-switched English sentence and the

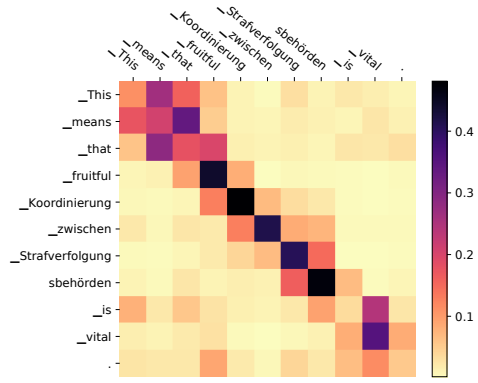


Figure 3: Visualization for the self-attention weights.

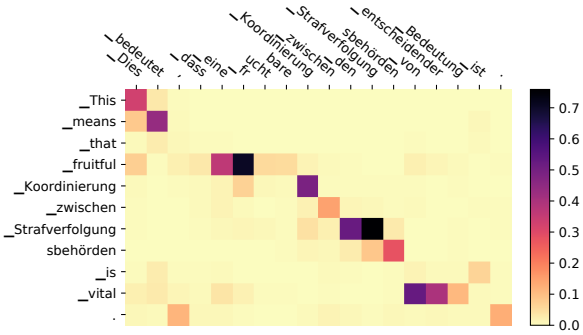
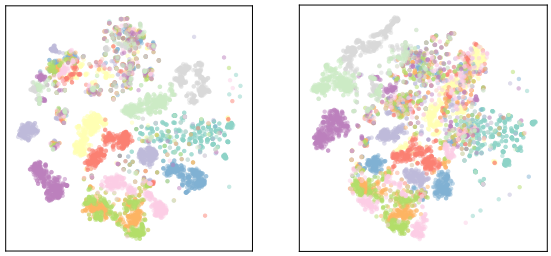


Figure 4: Visualization for the cross-attention weights between the code-switched input and target sentence.

output German sentence. The words with similar meanings are aligned together between the code-switched input and target output.

Crosslingual Representation We select 500 parallel sentences across different languages and visualize their sentence vectors of multilingual baseline and our method in Figure 5. The vector of the special language symbol of the source sentence is used as the sentence representation for visualization. Compared to Figure 5(a), different languages become closer and overlap with each other in Figure 5(b), which shows our method aligns representations and minimizes the differences among different languages.

Substitution Strategy We employ both word-level and phrase-level substitution strategies for code-switching. The word-level and phrase-level methods replace some words or spans of the source sentence with other languages. In Table 4, phrase-level substitution works better. Furthermore, we investigate the effect of the substitution ratio of the source words. From Figure 6, the best substitution ratio is 10%. When increasing the ratio to 30%, the performance gets worse, which indicates substitut-



(a) (b)

Figure 5: t-SNE (Maaten and Hinton, 2008) visualization of parallel sentences vector space of all languages from the multilingual baseline (a) and our method (b). Each color denotes one language.

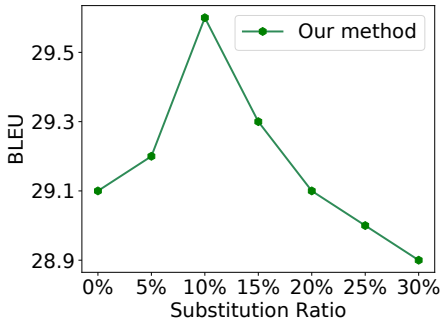


Figure 6: Average results of X→En directions on different substitution ratio settings. Large substitution ratio may degrade the model performance and is even worse than the multilingual baseline.

X → En	De	Lv	Ro	Tr	Avg
Word-level	42.5	21.5	35.9	26.2	31.5
Phrase-level	42.6	21.8	36.4	26.7	31.9

Table 4: Comparison of BLEU points between the word-level and the phrase-level substitution strategies on X→En directions.

ing too many words may degrade the performance.

As Equation 3 formulates, our method uses both the original corpora and code-switched corpora simultaneously to reduce the effect of the word alignment errors. Besides, `fast_align` (Dyer et al., 2013) is a simple, fast, and effective tool with a lower alignment error rate. Therefore, our method can avoid the disturbance introduced by the word alignment errors as much as possible.

Time Cost of Word Alignment In this work, we try a large pseudo parallel corpus (33M) to train the multilingual corpora. In most scenarios, the size of the parallel corpus is less than 33M and thus consumes less time to generate the alignment pairs. All the alignment pairs are offline generated only once before the training phase. Therefore, the time cost of the word alignment is much smaller than that of the model training.

6 Related Work

Multilingual Machine Translation Previous works (Zoph et al., 2016; Firat et al., 2016b; Johnson et al., 2017) have explored different settings of the multilingual neural machine translation (MNMT). Recent studies show that MNMT (Blackwood et al., 2018; Platanios et al., 2018; Gu et al., 2018) helps improve the performance of the low-resource or zero-shot translation. Some researchers use the sentence pairs to enhance the bilingual neural machine translation (Conneau and Lample, 2019; Song et al., 2019; Yang et al., 2020b).

Agreement-based Learning Many works try to use the agreement-based method (Liang et al., 2007, 2006; Al-Shedivat and Parikh, 2019) to encourage agreement among different translation orders and directions (Liang et al., 2006; Castilho, 2020; Yang et al., 2020a; Cheng et al., 2016; Zhang et al., 2019). Besides, the agreement-based method is also used to minimize the difference between the representation of source and target sentence (Yang et al., 2019). Our method further explores the approach of the multilingual agreement.

7 Conclusion

We propose a novel agreement-based framework to encourage multilingual agreement across different translation directions by the agreement term. Experimental results on the multilingual translation task demonstrate that our method effectively minimizes the gaps among different translation directions and significantly outperforms the multilingual baselines. The analytic experiment about the crosslingual representation shows the effectiveness of our multilingual agreement in minimizing the differences among different languages.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081, 61370126), the 2020 Tencent WeChat Rhino-Bird Focused Research Program, and the Fund of the State Key Laboratory of Software Development Environment (Grant No.SKLSDE2019ZX-17).

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *NAACL 2019*, pages 3874–3884.

Maruan Al-Shedivat and Ankur P. Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *NAACL 2019*, pages 1184–1197.

Graeme W. Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *COLING 2018*, pages 3112–3122.

Sheila Castilho. 2020. Document-level machine translation evaluation project: Methodology, effort and inter-annotator agreement. In *EAMT 2020*, pages 455–456.

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Agreement-based joint training for bidirectional attention-based neural machine translation. In *IJCAI 2016*, pages 2761–2767.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS 2019*, pages 7057–7067.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *NAACL 2013*, pages 644–648.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL 2016*, pages 866–875.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman-Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *EMNLP 2016*, pages 268–277.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *NAACL 2018*, pages 344–354.

Michael A. Hedderich, David Ifeoluwa Adelani, Dawei Zhu, Jesujoba O. Alabi, Udia Markus, and Dietrich Klakow. 2020. Transfer learning and distant supervision for multilingual transformer models: A study on african languages. In *EMNLP 2020*, pages 2580–2591.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP 2018*, pages 66–71.

Percy Liang, Dan Klein, and Michael I. Jordan. 2007. Agreement-based learning. In *NIPS 2007*, pages 913–920.

Percy Liang, Benjamin Taskar, and Dan Klein. 2006. Alignment by agreement. In *NAACL 2006*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom M. Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *EMNLP 2018*, pages 425–435.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT 2018*, pages 186–191.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *IJCAI 2020*, pages 3853–3860.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:140:1–140:67.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multilingual unsupervised NMT using shared encoder and language-specific decoders. In *ACL 2019*, pages 3083–3089.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *NAACL 2019*, pages 449–459.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *ACL 2020*, pages 3525–3535.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*, pages 5998–6008.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *ICLR 2019*.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *EMNLP 2020*, pages 1022–1034.

Mingming Yang, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Min Zhang, and Tiejun Zhao. 2019. Sentence-level agreement for neural machine translation. In *ACL 2019*, pages 3076–3082.

Mingming Yang, Xing Wang, Min Zhang, and Tiejun Zhao. 2020a. Incorporating phrase-level agreement into neural machine translation. In *NLPCC 2020*, pages 416–428.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020b. CSP: code-switching pre-training for neural machine translation. In *EMNLP 2020*, pages 2624–2636.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL 2020*, pages 1628–1639.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019. Regularizing neural machine translation by target-bidirectional agreement. In *AAAI 2019*, pages 443–450.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *EMNLP 2016*, pages 1568–1575.

Higher-order Derivatives of Weighted Finite-state Machines

Ran Zmigrod [1] **Tim Vieira** [2] **Ryan Cotterell** [1,3]
[1] University of Cambridge [2] Johns Hopkins University [3] ETH Zürich
rz279@cam.ac.uk tim.f.vieira@gmail.com
ryan.cotterell@inf.ethz.ch

Abstract

Weighted finite-state machines are a fundamental building block of NLP systems. They have withstood the test of time—from their early use in noisy channel models in the 1990s up to modern-day neurally parameterized conditional random fields. This work examines the computation of higher-order derivatives with respect to the normalization constant for weighted finite-state machines. We provide a general algorithm for evaluating derivatives of all orders, which has not been previously described in the literature. In the case of second-order derivatives, our scheme runs in the *optimal* $\mathcal{O}(A^2N^4)$ time where A is the alphabet size and N is the number of states. Our algorithm is significantly faster than prior algorithms. Additionally, our approach leads to a significantly faster algorithm for computing second-order expectations, such as covariance matrices and gradients of first-order expectations.

1 Introduction

Weighted finite-state machines (WFSMs) have a storied role in NLP. They are a useful formalism for speech recognition (Mohri et al., 2002), machine transliteration (Knight and Graehl, 1998), morphology (Geyken and Hanneforth, 2005; Lindén et al., 2009) and phonology (Cotterell et al., 2015) *inter alia*. Indeed, WFSMs have been "neuralized" (Rastogi et al., 2016; Hannun et al., 2020; Schwartz et al., 2018) and are still of practical use to the NLP modeler. Moreover, many popular sequence models, e.g., conditional random fields for part-of-speech tagging (Lafferty et al., 2001), are naturally viewed as special cases of WFSMs. For this reason, we consider the study of algorithms for the WFSMs of interest *in se* for the NLP community.

This paper considers inference algorithms for WSFMs. When WFSMs are acyclic, there exist simple linear-time dynamic programs, e.g., the forward algorithm (Rabiner, 1989), for inference. However, in general, WFSMs may contain cycles and such approaches are not applicable. Our work considers this general case and provides a method for efficient computation of m^{th}-order derivatives over a cyclic WFSM. To the best of our knowledge, no algorithm for higher-order derivatives has been presented in the literature beyond a general-purpose method from automatic differentiation. In contrast to many presentations of WFSMs (Mohri, 1997), our work provides a purely linear-algebraic take on them. And, indeed, it is this connection that allows us to develop our general algorithm.

We provide a thorough analysis of the soundness, runtime, and space complexity of our algorithm. In the special case of second-order derivatives, our algorithm runs *optimally* in $\mathcal{O}(A^2N^4)$ time and space where A is the size of the alphabet, and N is the number of states.[1] In contrast, the second-order expectation semiring of Li and Eisner (2009) provides an $\mathcal{O}(A^2N^7)$ solution and automatic differentiation (Griewank, 1989) yields a slightly faster $\mathcal{O}(AN^5{+}A^2N^4)$ solution. Additionally, we provide a speed-up for the general family of second-order expectations. Indeed, we believe our algorithm is the fastest known for computing common quantities, e.g., a covariance matrix.[2]

2 Weighted Finite-State Machines

In this section we briefly provide important notation for WFSMs and a classic result that efficiently finds the normalization constant for the probability distribution of a WFSM.

[1] Our implementation is available at https://github.com/rycolab/wfsm.

[2] Due to space constraints, we keep the discussion of our paper theoretical, though applications of expectations that we can compute are discussed in Li and Eisner (2009), Sánchez and Romero (2020), and Zmigrod et al. (2021).

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 240–248
August 1–6, 2021. ©2021 Association for Computational Linguistics

Definition 1. *A **weighted finite-state machine** $\mathcal{M}$ is a tuple $\langle \boldsymbol{\alpha}, \{\mathbf{W}^{(a)}\}_{a \in \overline{\mathcal{A}}}, \boldsymbol{\omega} \rangle$ where $\mathcal{A}$ is an alphabet of size A, $\overline{\mathcal{A}} \stackrel{\text{def}}{=} \mathcal{A} \cup \{\varepsilon\}$, each $a \in \overline{\mathcal{A}}$ has a symbol-specific transition matrix $\mathbf{W}^{(a)} \in {\mathbb{R}_{\geq 0}}^{N \times N}$ where N is the number of states, and $\boldsymbol{\alpha}, \boldsymbol{\omega} \in {\mathbb{R}_{\geq 0}}^{N}$ are column vectors of start and end weights, respectively. We define the matrix $\mathbf{W} \stackrel{\text{def}}{=} \sum_{a \in \overline{\mathcal{A}}} \mathbf{W}^{(a)}$.*

Definition 2. *A **trajectory** $\tau_{i \rightsquigarrow \ell}$ is an ordered sequence of transitions from state i to state ℓ. Visually, we can represent a trajectory by*

$$\tau_{i \rightsquigarrow \ell} \stackrel{\text{def}}{=} i \xrightarrow{a} j \cdots k \xrightarrow{a'} \ell$$

*The **weight** of a trajectory is*

$$w(\tau_{i \rightsquigarrow \ell}) \stackrel{\text{def}}{=} \alpha_i \left(\prod_{(j \xrightarrow{a} k) \in \tau_{i \rightsquigarrow \ell}} \mathrm{W}^{(a)}_{jk} \right) \omega_\ell \tag{1}$$

We denote the (possibly infinite) set of trajectories from i to ℓ by $\mathcal{T}_{i\ell}$ and the set of all trajectories by $\mathcal{T} \stackrel{\text{def}}{=} \bigcup_{i,\ell \in [N]} \mathcal{T}_{i\ell}$.[3] Consequently, when we say $\tau_{i \rightsquigarrow \ell} \in \mathcal{T}$, we make i and ℓ implicit arguments to which $\mathcal{T}_{i\ell}$ we are accessing.

We define the **probability** of a trajectory $\tau_{i \rightsquigarrow \ell} \in \mathcal{T}$,

$$p(\tau_{i \rightsquigarrow \ell}) \stackrel{\text{def}}{=} \frac{w(\tau_{i \rightsquigarrow \ell})}{\mathrm{Z}} \tag{2}$$

where

$$\mathrm{Z} \stackrel{\text{def}}{=} \boldsymbol{\alpha}^\top \sum_{k=0}^{\infty} \mathbf{W}^k \, \boldsymbol{\omega} \tag{3}$$

Of course, p is only well-defined when $0 < \mathrm{Z} < \infty$.[4] Intuitively, $\boldsymbol{\alpha}^\top \mathbf{W}^k \, \boldsymbol{\omega}$ is the total weight of all trajectories of length k. Thus, Z is the total weight of all possible trajectories as it sums over the total weight for each possible trajectory length.

Theorem 1 (Corollary 4.2, Lehmann (1977))**.**

$$\mathbf{W}^\star \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \mathbf{W}^k = (\mathbf{I} - \mathbf{W})^{-1} \tag{4}$$

Thus, we can solve the infinite summation that defines $\mathbf{W}^\star$ by matrix inversion in $\mathcal{O}(N^3)$ time.[5]

[3] $|\mathcal{T}|$ is infinite if and only if $\mathcal{M}$ is cyclic.

[3] Another formulation for Z is $\sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}} w(\tau_{i \rightsquigarrow \ell})$.

[4] This requirement is equivalent to $\mathbf{W}$ having a spectral radius < 1.

[5] This solution technique may be extended to closed semirings (Kleene, 1956; Lehmann, 1977).

Corollary 1.

$$\mathrm{Z} = \boldsymbol{\alpha}^\top \mathbf{W}^\star \, \boldsymbol{\omega} \tag{5}$$

Proof. Follows from (4) in Theorem 1. ■

By Corollary 1, we can find Z in $\mathcal{O}(N^3 + AN^2)$.[6]

Strings versus Trajectories. Importantly, WFSMs can be regarded as weighted finite-state acceptors (WFSAs) which accept strings as their input. Each trajectory $\tau_{i \rightsquigarrow \ell} \in \mathcal{T}$ has a **yield** $\gamma(\tau_{i \rightsquigarrow \ell})$ which is the concatenation of the alphabet symbols of the trajectory. The yield of a trajectory ignores any ε symbols, a discussion regarding the semantics of ε is given in Hopcroft et al. (2001). As we focus on distributions over trajectories, we do not need special considerations for ε transitions. We do not consider distributions over yields in this work as such a distribution requires constructing a latent-variable model

$$p(\sigma) = \frac{1}{\mathrm{Z}} \sum_{\substack{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}, \\ \gamma(\tau_{i \rightsquigarrow \ell}) = \sigma}} w(\tau_{i \rightsquigarrow \ell}) \tag{6}$$

where $\sigma \in \mathcal{A}^*$ and $\gamma(\tau_{i \rightsquigarrow \ell})$ is the yield of the trajectory. While marginal likelihood can be found efficiently,[7] many quantities, such as the entropy of the distribution over yields, are intractable to compute (Cortes et al., 2008).

3 Computing the Hessian (and Beyond)

In this section, we explore algorithms for efficiently computing the Hessian matrix $\nabla^2 \mathrm{Z}$. We briefly describe two inefficient algorithms, which are derived by forward-mode and reverse-mode automatic differentiation. Next, we propose an efficient algorithm which is based on a key differential identity.

3.1 An $\mathcal{O}(A^2 N^7)$ Algorithm with Forward-Mode Automatic Differentiation

One proposal for computing the Hessian comes from Li and Eisner (2009) who introduce a method based on semirings for computing a general family of quantities known as second-order expectations

[6] Throughout this paper, we assume a dense weight matrix and that matrix inversion is $\mathcal{O}(N^3)$ time. We note, however, that when the weight matrix is sparse and structured, faster matrix-inversion algorithms exist that exploit the strongly connected components decomposition of the graph (Mohri et al., 2000). We are agnostic to the specific inversion algorithm, but for simplicity we assume the aforementioned running time.

[7] This is done by intersecting the WFSA with another WFSA that only accepts σ.

(defined formally in §4). When applied to the computation of the Hessian their method reduces precisely to forward-mode automatic differentiation (AD; Griewank and Walther, 2008, Chap 3.1). This approach requires that we "lift" the computation of Z to operate over a richer numeric representation known as *dual numbers* (Clifford, 1871; Pearlmutter and Siskind, 2007). Unfortunately, the second-order dual numbers that we require to compute the Hessian introduce an overhead of $\mathcal{O}(A^2N^4)$ per numeric operation of the $\mathcal{O}(N^3)$ algorithm that computes Z, which results in $\mathcal{O}(A^2N^7)$ time.

3.2 An $\mathcal{O}(AN^5+A^2N^4)$ Algorithm with Reverse-Mode Automatic Differentiation

Another method for materializing the Hessian $\nabla^2\mathrm{Z}$ is through reverse-mode automatic differentiation (AD). Recall that we can compute Z in $\mathcal{O}(N^3+AN^2)$, and can consequently find $\nabla\mathrm{Z}$ in $\mathcal{O}(N^3+AN^2)$ using one pass of reverse-mode AD (Griewank and Walther, 2008, Chapter 3.3). We can repeat differentiation to materialize $\nabla^2\mathrm{Z}$. Specifically, we run reverse-mode AD once for each element i of $\nabla\mathrm{Z}$. Taking the gradient of $(\nabla\mathrm{Z})_i$ gives a row of the Hessian matrix, $\nabla[(\nabla\mathrm{Z})_i] = [\nabla^2\mathrm{Z}]_{(i,:)}$. Since each of these passes takes time $\mathcal{O}(N^3+AN^2)$ (i.e., the same as the cost of Z), and $\nabla\mathrm{Z}$ has size AN^2, the overall time is $\mathcal{O}(AN^5+A^2N^4)$.

3.3 Our Optimal $\mathcal{O}(A^2N^4)$ Algorithm

In this section, we will provide an $\mathcal{O}(A^2N^4)$-time and space algorithm for computing the Hessian. Since the Hessian has size $\mathcal{O}(A^2N^4)$, no algorithm can run faster than this bound; thus, our algorithm's time and space complexities are *optimal*. Our algorithm hinges on the following lemma, which shows that the each of partial derivatives of $\mathbf{W}^\star$ can be cheaply computed given $\mathbf{W}^\star$.

Lemma 1. *For $i,j,k,\ell \in [N]$ and $a \in \overline{\mathcal{A}}$*

$$\frac{\partial \mathrm{W}^\star_{i\ell}}{\partial \mathrm{W}^{(a)}_{jk}} = \mathrm{W}^\star_{ij}\dot{\mathrm{W}}^{(a)}_{jk}\mathrm{W}^\star_{k\ell} \tag{7}$$

where $\dot{\mathrm{W}}^{(a)}_{jk}$ is shorthand for $\partial \mathrm{W}^{(a)}_{jk}$.

Proof.

$$\begin{aligned}\frac{\partial \mathrm{W}^\star_{i\ell}}{\partial \mathrm{W}^{(a)}_{jk}} &= \frac{\partial}{\partial \mathrm{W}^{(a)}_{jk}}\left[(\mathbf{I}-\mathbf{W})^{-1}_{i\ell}\right] \\ &= -\mathrm{W}^\star_{ij}\frac{\partial}{\partial \mathrm{W}^{(a)}_{jk}}\left[(\mathbf{I}-\mathbf{W})\right]\mathrm{W}^\star_{k\ell} \\ &= \mathrm{W}^\star_{ij}\dot{\mathrm{W}}^{(a)}_{jk}\mathrm{W}^\star_{k\ell}\end{aligned}$$

The second step uses Equation 40 of the Matrix Cookbook (Petersen and Pedersen, 2008). ■

We now extend Lemma 1 to express higher-order derivatives in terms of $\mathbf{W}^\star$. Note that as in Lemma 1, we will use $\dot{\mathrm{W}}^{(a)}_{ij}$ as a shorthand for the partial derivative $\partial \mathrm{W}^{(a)}_{ij}$.

Theorem 2. *For $m \geq 1$ and m-tuple of transitions $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1, \ldots, i_m \xrightarrow{a_m} j_m\rangle$*

$$\frac{\partial^m \mathrm{Z}}{\partial \mathrm{W}^{(a_1)}_{i_1j_1}\cdots\partial \mathrm{W}^{(a_m)}_{i_mj_m}} = \sum_{\left\langle i'_1 \xrightarrow{a'_1} j'_1,\cdots,i'_m \xrightarrow{a'_m} j'_m\right\rangle \in \mathcal{S}_{\vec{\tau}}} \mathrm{s}_{i'_1}\dot{\mathrm{W}}^{(a'_1)}_{i'_1j'_1}\mathrm{W}^\star_{j'_1i'_2}\dot{\mathrm{W}}^{(a'_2)}_{i'_2j'_2}\cdots\mathrm{W}^\star_{j'_{m-1}i'_m}\dot{\mathrm{W}}^{(a'_m)}_{i'_mj'_m}\mathrm{e}_{j'_m} \tag{8}$$

where $\mathbf{s} = \boldsymbol{\alpha}^\top\mathbf{W}^\star$, $\mathbf{e} = \mathbf{W}^\star\boldsymbol{\omega}$ and $\mathcal{S}_{\vec{\tau}}$ is the multi-set of permutations of $\vec{\tau}$.[8]

Proof. See App. A.1 ■

Corollary 2. *For $i,j,k,l \in [N]$ and $a,b \in \overline{\mathcal{A}}$*

$$\frac{\partial^2 \mathrm{Z}}{\partial \mathrm{W}^{(a)}_{ij}\partial \mathrm{W}^{(b)}_{kl}} = \mathrm{s}_i\dot{\mathrm{W}}^{(a)}_{ij}\mathrm{W}^\star_{jk}\dot{\mathrm{W}}^{(b)}_{kl}\mathrm{e}_l + \mathrm{s}_k\dot{\mathrm{W}}^{(b)}_{kl}\mathrm{W}^\star_{li}\dot{\mathrm{W}}^{(a)}_{ij}\mathrm{e}_j \tag{9}$$

Proof. Application of Theorem 2 for the m=2 case. ■

Theorem 2 shows that, if we have already computed $\mathbf{W}^\star$, each element of the m^{th} derivative can be found in $\mathcal{O}(m\,m!)$ time: We must sum over $\mathcal{O}(m!)$ permutations, where each summand is the product of $\mathcal{O}(m)$ items. Importantly, for the Hessian ($m=2$), we can find each element in $\mathcal{O}(1)$ using Corollary 2. Algorithm $\mathtt{D}_m$ in Fig. 1 provides pseudocode for materializing the tensor containing the m^{th} derivatives of Z.

Theorem 3. *For $m \geq 1$, algorithm $\mathtt{D}_m$ computes $\nabla^m\mathrm{Z}$ in $\mathcal{O}(N^3 + m\,m!\,A^mN^{2m})$ time and $\mathcal{O}(A^mN^{2m})$ space.*

Proof. Correctness of algorithm $\mathtt{D}_m$ follows from Theorem 2. The runtime and space bounds follow by needing to compute and store each combination of transitions. Each line of the algorithm is annotated with its running time. ■

Corollary 3. *The Hessian $\nabla^2\mathrm{Z}$ can be materialized in $\mathcal{O}(A^2N^4)$ time and $\mathcal{O}(A^2N^4)$ space. Note that these bounds are optimal.*

[8] As $\vec{\tau}$ may have duplicates, $\mathcal{S}_{\vec{\tau}}$ can also have duplicates and so must be a multi-set.

def $\mathtt{D}_m(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\omega})$:

▷ *Compute the tensor of m^{th}-order derivative of a WFSM; requires $\mathcal{O}(N^3 + m\, m!\, A^m N^{2m})$ time, $\mathcal{O}(A^m N^{2m})$ space.*

$\mathbf{W}^\star \leftarrow (\mathbf{I} - \mathbf{W})^{-1}$ ▷ $\mathcal{O}(N^3)$

$\mathbf{s} \leftarrow \boldsymbol{\alpha}^\top \mathbf{W}^\star$; $\mathbf{e} \leftarrow \mathbf{W}^\star \boldsymbol{\omega}$ ▷ $\mathcal{O}(N^2)$

$\mathbf{D} \leftarrow \mathbf{0}$

for $\vec{\tau} \in ([N] \times [N] \times \overline{\mathcal{A}})^m$: ▷ $\mathcal{O}(mm!A^m N^{2m})$

  for $\langle i_1 \xrightarrow{a_1} j_1, \ldots, i_m \xrightarrow{a_m} j_m \rangle \in \mathcal{S}_{\vec{\tau}}$:

$$\mathrm{D}_{\vec{\tau}} \mathrel{+}= \mathrm{s}_{i_1} \dot{\mathrm{W}}^{(a_1)}_{i_1 j_1} \mathrm{W}^\star_{j_1 i_2} \dot{\mathrm{W}}^{(a_2)}_{i_2 j_2} \mathrm{W}^\star_{j_2 i_3} \cdots \mathrm{W}^\star_{j_{m-1} i_m} \dot{\mathrm{W}}^{(a_m)}_{i_m j_m} \mathrm{e}_{j_m}$$

return $\mathbf{D}$

def $\mathtt{E}_2(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\omega}, r, t)$:

▷ *Compute the second-order expectation of a WFSM; requires $\mathcal{O}(N^3 + N^2(\overline{R}\,\overline{T} + AR'T'))$ time, $\mathcal{O}(N^2 + RT + N(R+T))$ space where $\overline{R} \stackrel{\text{def}}{=} \min(NR', R)$ and $\overline{T} \stackrel{\text{def}}{=} \min(NT', T)$.*

Compute $\mathbf{W}^\star$, $\mathbf{s}$, and $\mathbf{e}$ as in $\mathtt{D}_m$ ▷ $\mathcal{O}(N^3)$

$\mathrm{Z} \leftarrow \boldsymbol{\alpha}^\top \mathbf{W}^\star \boldsymbol{\omega}$

$\widehat{r^s} \leftarrow \mathbf{0}$; $\widehat{r^e} \leftarrow \mathbf{0}$; $\widehat{t^s} \leftarrow \mathbf{0}$; $\widehat{t^e} \leftarrow \mathbf{0}$

for $i, j \in [N], a \in \overline{\mathcal{A}}$: ▷ $\mathcal{O}(AN^2)$

$\widehat{r^s_i}$ += $\mathrm{s}_j \dot{\mathrm{W}}^{(a)}_{ji} \mathrm{W}^{(a)}_{ji} r^{(a)}_{ji}$ ▷ $\mathcal{O}(R')$

$\widehat{r^e_i}$ += $\dot{\mathrm{W}}^{(a)}_{ij} \mathrm{e}_j \dot{\mathrm{W}}^{(a)}_{ji} \mathrm{W}^{(a)}_{ij} r^{(a)}_{ij}$ ▷ $\mathcal{O}(R')$

$\widehat{t^s_i}$ += $\mathrm{s}_j \dot{\mathrm{W}}^{(i)}_{aj} \mathrm{W}^{(a)}_{ji} t^{(a)}_{ji}$ ▷ $\mathcal{O}(T')$

$\widehat{t^e_i}$ += $\dot{\mathrm{W}}^{(a)}_{ij} \mathrm{e}_j \mathrm{W}^{(a)}_{ij} t^{(a)}_{ij}$ ▷ $\mathcal{O}(T')$

return $\frac{1}{\mathrm{Z}} \Big[\sum_{i,j=0}^{N} \widehat{r^s_i} \mathrm{W}^\star_{ij} \widehat{t^e_j}^\top + \Big[\widehat{t^s_i} \mathrm{W}^\star_{ij} \widehat{r^e_j}^\top \Big]^\top + \sum_{a \in \overline{\mathcal{A}}} \mathrm{s}_i \dot{\mathrm{W}}^{(a)}_{ij} \mathrm{e}_j \mathrm{W}^{(a)}_{ij} r^{(a)}_{ij} t^{(a)\top}_{ij} \Big]$

▷ $\mathcal{O}(N^2(\overline{R}\,\overline{T} + AR'T'))$

Figure 1: Algorithms

Proof. Application of Theorem 3 for the m=2 case. ■

4 Second-Order Expectations

In this section, we leverage the algorithms of the previous section to efficiently compute a family expectations, known as a second-order expectations. To begin, we define an **additively decomposable** function $r\colon \mathcal{T} \mapsto \mathbb{R}^R$ as any function expressed as

$$r(\tau_{i \rightsquigarrow \ell}) = \sum_{(j \xrightarrow{a} k) \in \tau_{i \rightsquigarrow \ell}} r^{(a)}_{jk} \tag{10}$$

where each $r^{(a)}_{jk}$ is an R-dimensional vector. Since many r of interest are sparse, we analyze our algorithms in terms of R and its maximum density $R' \stackrel{\text{def}}{=} \max_{j \xrightarrow{a} k} \|r^{(a)}_{jk}\|_0$. Previous work has considered expectations of such functions (Eisner, 2001) and the *product* of two such functions (Li and Eisner, 2009), better known as second-order expectations. Formally, given two additively decomposable functions $r\colon \mathcal{T} \mapsto \mathbb{R}^R$ and $t\colon \mathcal{T} \mapsto \mathbb{R}^T$, a **second-order expectation** is

$$\mathbb{E}_{\tau_{i \rightsquigarrow \ell}} \Big[r(\tau_{i \rightsquigarrow \ell}) t(\tau_{i \rightsquigarrow \ell})^\top \Big] \stackrel{\text{def}}{=} \sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}} p(\tau_{i \rightsquigarrow \ell}) r(\tau_{i \rightsquigarrow \ell}) t(\tau_{i \rightsquigarrow \ell})^\top \tag{11}$$

Examples of second-order expectations include the Fisher information matrix and the gradients of first-order expectations (e.g., expected cost, entropy, and the Kullback–Leibler divergence).

Our algorithm is based on two fundamental concepts. Firstly, expectations for probability distributions as described in (1), can be decomposed as expectations over transitions (Zmigrod et al., 2021). Secondly, the marginal probabilities of transitions are connected to derivatives of Z.[9]

Lemma 2. *For $m \geq 1$ and m-tuple of transitions $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1, \ldots, i_m \xrightarrow{a_m} j_m \rangle$*

$$p(\vec{\tau}) = \frac{1}{\mathrm{Z}} \sum_{n=1}^{m} \frac{\partial^n \mathrm{Z}}{\partial \mathrm{W}^{(a_1)}_{i_1 j_1} \ldots \partial \mathrm{W}^{(a_n)}_{i_n j_n}} \prod_{k=1}^{n} \mathrm{W}^{(a_k)}_{i_k j_k} \tag{12}$$

Proof. See App. A.2. ■

We formalize our algorithm as $\mathtt{E}_2$ in Fig. 1. Note that we achieve an additional speed-up by exploiting associativity (see App. A.3).

Theorem 4. *Algorithm $\mathtt{E}_2$ computes the second-order expectation of additively decomposable functions $r\colon \mathcal{T} \mapsto \mathbb{R}^R$ and $t\colon \mathcal{T} \mapsto \mathbb{R}^T$ in:*

$$\mathcal{O}(N^3 + N^2(\overline{R}\,\overline{T} + AR'T')) \text{ time}$$
$$\mathcal{O}(N^2 + RT + N(R+T)) \text{ space}$$

where $\overline{R} = \min(NR', R)$ and $\overline{T} = \min(NT', T)$.

Proof. Correctness of algorithm $\mathtt{E}_2$ is given in App. A.3. The runtime bounds are annotated on each line of the algorithm. We note that each $\widehat{r}$ and $\widehat{t}$ is $\overline{R}$ and $\overline{T}$ sparse. $\mathcal{O}(N^2)$ space is required to store $\mathbf{W}^\star$, $\mathcal{O}(RT)$ is required to store the expectation, and $\mathcal{O}(N(R+T))$ space is required to store the various $\widehat{r}$ and $\widehat{t}$ quantities. ■

Previous approaches for computing second-order expectations are significantly slower than $\mathtt{E}_2$. Specifically, using Li and Eisner (2009)'s second-order expectation semiring requires augmenting the

[9] This is commonly used in the case of single transition marginals, which can be found by $\nabla \log \mathrm{Z}$

arc weights to be $R \times T$ matrices and so runs in $\mathcal{O}(N^3RT + AN^2RT)$. Alternatively, we can use AD, as in §3.2, to materialize the Hessian and compute the pairwise transition marginals. This would result in a total runtime of $\mathcal{O}(AN^5 + A^2N^4R'T')$.

5 Conclusion

We have presented efficient methods that exploit properties of the derivative of a matrix inverse to find m-order derivatives for WFSMs. Additionally, we provided an explicit, novel, algorithm for materializing the Hessian in its *optimal* complexity, $\mathcal{O}(A^2N^4)$. We also showed how this could be utilized to efficiently compute second-order expectations of distributions under WFSMs, such as covariance matrices and the gradient of entropy. We hope that our paper encourages future research to use the Hessian and second-order expectations of WFSM systems, which have previously been disadvantaged by inefficient algorithms.

Acknowledgments

We would like to thank the reviewers for engagine with our work and providing valuable feedback. The first author is supported by the University of Cambridge School of Technology Vice-Chancellor's Scholarship as well as by the University of Cambridge Department of Computer Science and Technology's EPSRC.

Ethical Concerns

We do not foresee how the more efficient algorithms presented this work exacerbate any existing ethical concerns with NLP systems.

References

W. K. Clifford. 1871. Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, 1.

Corinna Cortes, Mehryar Mohri, Ashish Rastogi, and Michael Riley. 2008. On the computation of the relative entropy of probabilistic automata. *International Journal of Foundations of Computer Science*, 19.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3.

Jason Eisner. 2001. Expectation semirings: Flexible EM for learning finite-state transducers. In *Proceedings of the European Summer School in Logic, Language and Information Workshop on Finite-state Methods in Natural Language Processing*.

Alexander Geyken and Thomas Hanneforth. 2005. TAGH: A complete morphology for German based on weighted finite state automata. In *Finite-State Methods and Natural Language Processing, 5th International Workshop*.

Andreas Griewank. 1989. On automatic differentiation. *Mathematical Programming: Recent Developments and Applications*, 6.

Andreas Griewank and Andrea Walther. 2008. *Evaluating Derivatives–Principles and Techniques of Algorithmic Differentiation*, 2nd edition. Society for Industrial and Applied Mathematics.

Awni Hannun, Vineel Pratap, Jacob Kahn, and Wei-Ning Hsu. 2020. Differentiable weighted finite-state transducers. *CoRR*, abs/2010.01003.

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. 2001. *Introduction to automata theory, languages, and computation, 2nd Edition*. Addison-Wesley series in computer science. Addison-Wesley-Longman.

Stephen C. Kleene. 1956. Representation of events in nerve nets and finite automata. *Automata Studies*.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Daniel J. Lehmann. 1977. Algebraic structures for transitive closure. *Theoretical Computer Science*, 4.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Krister Lindén, Miikka Silfverberg, and Tommi A. Pirinen. 2009. HFST tools for morphology - an efficient open-source package for construction of morphological analyzers. In *Proceedings of the State of the Art in Computational Morphology - Workshop on Systems and Frameworks for Computational Morphology*.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 2000. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231.

Barak A. Pearlmutter and Jeffrey Mark Siskind. 2007. Lazy multivariate higher-order forward-mode AD. In *Proceedings of the 34th Association for Computer Machinery Special Interest Group on Programming Languages and Special Interest Group on Algorithms and Computation Theory Symposium on Principles of Programming Languages*.

K. B. Petersen and M. S. Pedersen. 2008. The matrix cookbook. Version 20081110.

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the Institute of Electrical and Electronics Engineers*, 77.

Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. Weighting finite-state transductions with neural context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Joan-Andreu Sánchez and Verónica Romero. 2020. Computation of moments for probabilistic finite-state automata. *Information Sciences*, 516.

Roy Schwartz, Sam Thomson, and Noah A. Smith. 2018. Bridging CNNs, RNNs, and weighted finite-state machines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1.

Ran Zmigrod, Tim Vieira, and Ryan Cotterell. 2021. Efficient computation of expectations under spanning tree distributions. *Transactions of the Association for Computational Linguistics*.

A Proofs

A.1

Theorem 2. *For $m \geq 1$ and m-tuple of transitions $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1, \ldots, i_m \xrightarrow{a_m} j_m \rangle$*

$$\frac{\partial^m \mathrm{Z}}{\partial \mathrm{W}^{(a_1)}_{i_1 j_1} \ldots \partial \mathrm{W}^{(a_m)}_{i_m j_m}} = \sum_{\left\langle i'_1 \xrightarrow{a'_1} j'_1, \ldots, i'_m \xrightarrow{a'_m} j'_m \right\rangle \in \mathcal{S}_{\vec{\tau}}} \mathrm{s}_{i'_1} \dot{\mathrm{W}}^{(a'_1)}_{i'_1 j'_1} \mathrm{W}^{\star}_{j'_1 i'_2} \dot{\mathrm{W}}^{(a'_2)}_{i'_2 j'_2} \cdots \mathrm{W}^{\star}_{j'_{m-1} i'_m} \dot{\mathrm{W}}^{(a'_m)}_{i'_m j'_m} \mathrm{e}_{j'_m}$$

where $\mathbf{s} = \boldsymbol{\alpha}^\top \mathbf{W}^\star$, $\mathbf{e} = \mathbf{W}^\star \boldsymbol{\omega}$ and $\mathcal{S}_{\vec{\tau}}$ is the multi-set of permutations of $\vec{\tau}$.

Proof. We prove this by induction on m.
Base Case: $m = 1$ and $\vec{\tau} = \langle i \xrightarrow{a} j \rangle$:

$$\frac{\partial \mathrm{Z}}{\partial \mathrm{W}^{(a)}_{ij}} = \frac{\partial}{\partial \mathrm{W}^{(a)}_{ij}} \left[\sum_{k,l=0}^{N} \alpha_k \mathrm{W}^{\star}_{kl} \omega_l \right] = \sum_{k,l=0}^{N} \alpha_k \mathrm{W}^{\star}_{ki} \dot{\mathrm{W}}^{(a)}_{ij} \mathrm{W}^{\star}_{jl} \omega_l = \mathrm{s}_i \dot{\mathrm{W}}^{(a)}_{ij} \mathrm{e}_j$$

Inductive Step: Assume that the expression holds for m. Let $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1, \ldots, i_m \xrightarrow{a_m} j_m \rangle$ and consider the tuple $\vec{\tau}\,'$, the concatenation of $(i \xrightarrow{a} j)$ and $\vec{\tau}$.

$$\frac{\partial^{m+1} \mathrm{Z}}{\mathrm{W}^{(a)}_{ij} \partial \mathrm{W}^{(a_1)}_{i_1 j_1} \ldots \partial \mathrm{W}^{(a_m)}_{i_m j_m}} = \frac{\partial}{\partial \mathrm{W}^{(a)}_{ij}} \sum_{\left\langle i'_1 \xrightarrow{a'_1} j'_1, \ldots, i'_m \xrightarrow{a'_m} j'_m \right\rangle \in \mathcal{S}_{\vec{\tau}}} \mathrm{s}_{i'_1} \dot{\mathrm{W}}^{(a'_1)}_{i'_1 j'_1} \mathrm{W}^{\star}_{j'_1 i'_2} \cdots \dot{\mathrm{W}}^{(a'_m)}_{i'_m j'_m} \mathrm{e}_{j'_m}$$

Consider the derivative of each summand with respect to $\mathrm{W}^{(a)}_{ij}$. By the product rule, we have

$$\begin{aligned}
&\frac{\partial}{\partial \mathrm{W}^{(a)}_{ij}} \left[\mathrm{s}_{i'_1} \dot{\mathrm{W}}^{(a'_1)}_{i'_1 j'_1} \mathrm{W}^{\star}_{j'_1 i'_2} \cdots \dot{\mathrm{W}}^{(a'_m)}_{i'_m j'_m} \mathrm{e}_{j'_m} \right] \\
&= \mathrm{s}_i \dot{\mathrm{W}}^{(a)}_{ij} \mathrm{W}^{\star}_{j i'_1} \dot{\mathrm{W}}^{(a'_1)}_{i'_1 j'_1} \mathrm{W}^{\star}_{j'_1 i'_2} \cdots \dot{\mathrm{W}}^{(a'_m)}_{i'_m j'_m} \mathrm{e}_{j'_m} + \\
&\quad \cdots + \mathrm{s}_{i'_1} \cdots \mathrm{W}^{\star}_{j_k i} \dot{\mathrm{W}}^{(a)}_{ij} \mathrm{W}^{\star}_{j i_{k+1}} \cdots \mathrm{e}_{j'_m} + \\
&\quad \cdots + \mathrm{s}_{i'_1} \dot{\mathrm{W}}^{(a'_1)}_{i'_1 j'_1} \mathrm{W}^{\star}_{j'_1 i'_2} \cdots \dot{\mathrm{W}}^{(a'_m)}_{i'_m j'_m} \mathrm{W}^{\star}_{j'_m i} \dot{\mathrm{W}}^{(a)}_{ij} \mathrm{e}_j
\end{aligned}$$

The above expression is equal to inserting $i \xrightarrow{a} j$ in every spot of the induction hypothesis's permutation, thereby creating a permutation over $\vec{\tau}\,'$. Reassembling with the expression for the derivative,

$$\frac{\partial^{m+1} \mathrm{Z}}{\partial \mathrm{W}^{(a)}_{ij} \partial \mathrm{W}^{(a_1)}_{i_1 j_1} \ldots \partial \mathrm{W}^{(a_m)}_{i_m j_m}} = \sum_{\left\langle i'_1 \xrightarrow{a'_1} j'_1, \ldots, i'_{m+1} \xrightarrow{a'_{m+1}} j'_{m+1} \right\rangle) \in \mathcal{S}_{\vec{\tau}\,'}} \mathrm{s}_{i'_1} \dot{\mathrm{W}}^{(a'_1)}_{i'_1 j'_1} \mathrm{W}^{\star}_{j'_1 i'_2} \dot{\mathrm{W}}^{(a'_2)}_{i'_2 j'_2} \cdots \dot{\mathrm{W}}^{(a'_{m+1})}_{i'_{m+1} j'_{m+1}} \mathrm{e}_{j'_{m+1}}$$

■

A.2

Lemma 2. *For $m \geq 1$ and m-tuple of transitions $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1, \ldots, i_m \xrightarrow{a_m} j_m \rangle$*

$$p(\vec{\tau}) = \frac{1}{\mathrm{Z}} \sum_{n=1}^{m} \frac{\partial^n \mathrm{Z}}{\partial \mathrm{W}^{(a_1)}_{i_1 j_1} \ldots \partial \mathrm{W}^{(a_n)}_{i_n j_n}} \prod_{k=1}^{n} \mathrm{W}^{(a_k)}_{i_k j_k} \tag{10}$$

Proof. Let $\mathcal{T}_{\vec{\tau}}$ be the set of trajectories such that $\tau_{i \rightsquigarrow \ell} \in \mathcal{T}_{\vec{\tau}} \iff \vec{\tau} \subseteq \tau_{i \rightsquigarrow \ell}$. Then,

$$p(\vec{\tau}) = \frac{1}{\mathrm{Z}} \sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}_{\vec{\tau}}} w(\tau_{i \rightsquigarrow \ell})$$

We prove the lemma by induction on m.
Base Case: Then, $m = 1$ and $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1 \rangle$. We have that

$$\frac{1}{\mathrm{Z}} \frac{\partial \mathrm{Z}}{\partial \mathrm{W}_{i_1 j_1}^{(a_1)}} \mathrm{W}_{i_1 j_1}^{(a_1)} = \frac{1}{\mathrm{Z}} \frac{\partial}{\partial \mathrm{W}_{i_1 j_1}^{(a_1)}} \left[\sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}} w(\tau_{i \rightsquigarrow \ell}) \right] \mathrm{W}_{i_1 j_1}^{(a_1)} \overset{\text{(a)}}{=} \frac{1}{\mathrm{Z}} \left(\sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}_{\vec{\tau}}} w(\tau_{i \rightsquigarrow \ell}) \right) = p(i_1 \xrightarrow{a_1} j_1)$$

Step (a) holds because taking the derivative of Z with respect to $\mathrm{W}_{i_1 j_1}^{(a_1)}$ yields the sum of the weights all trajectories which include $i_1 \xrightarrow{a_1} j_1$ where we exclude $\mathrm{W}_{i_1 j_1}^{(a_1)}$ from the computation of the weight. Then, we can push the outer $\mathrm{W}_{i_1 j_1}^{(a_1)}$ into the equation to obtain the sum of the weights of all trajectories containing $i_1 \xrightarrow{a_1} j_1$.
Inductive Step: Suppose that (10) holds for any m-tuple. Let $\vec{\tau} = \langle i_1 \xrightarrow{a_1} j_1, \ldots, i_{m+1} \xrightarrow{a_{m+1}} j_{m+1} \rangle$. Without loss of generality, fix $i_1 \xrightarrow{a_1} j_1$ and let $\vec{\tau}'$ be $\vec{\tau}$ without $i_1 \xrightarrow{a_1} j_1$.

$$\begin{aligned}
&\frac{1}{\mathrm{Z}} \sum_{n=1}^{m+1} \frac{\partial^n \mathrm{Z}}{\partial \mathrm{W}_{i_1 j_1}^{(a_1)} \ldots \partial \mathrm{W}_{i_n j_n}^{(a_n)}} \prod_{k=1}^{n} \mathrm{W}_{i_k j_k}^{(a_k)} \\
&\overset{\text{(b)}}{=} \mathrm{W}_{i_1 j_1}^{(a_1)} \frac{\partial}{\partial \mathrm{W}_{i_1 j_1}^{(a_1)}} \underbrace{\left[\frac{1}{\mathrm{Z}} \sum_{n=2}^{m+1} \frac{\partial^{(n-1)} \mathrm{Z}}{\partial \mathrm{W}_{i_2 j_2}^{(a_2)} \ldots \partial \mathrm{W}_{i_n j_n}^{(a_n)}} \prod_{k=2}^{n} \mathrm{W}_{i_k j_k}^{(a_k)} \right]}_{\text{Inductive hypothesis}} \\
&\overset{\text{(c)}}{=} \mathrm{W}_{i_1 j_1}^{(a_1)} \frac{\partial}{\partial \mathrm{W}_{i_1 j_1}^{(a_1)}} \overbrace{\left[\frac{1}{\mathrm{Z}} \sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}_{\vec{\tau}'}} w(\tau_{i \rightsquigarrow \ell}) \right]} \\
&\overset{\text{(d)}}{=} \frac{1}{\mathrm{Z}} \frac{\partial}{\partial \mathrm{W}_{i_1 j_1}^{(a_1)}} \left[\sum_{\tau_{i \rightsquigarrow \ell} \in \mathcal{T}_{\vec{\tau}'}} w(\tau_{i \rightsquigarrow \ell}) \right] \mathrm{W}_{i_1 j_1}^{(a_1)} \\
&\overset{\text{(e)}}{=} p(\vec{\tau})
\end{aligned}$$

Step (b) pushes $\frac{1}{\mathrm{Z}}$ and $\prod_{k=2}^{n} \mathrm{W}_{i_k j_k}^{(a_k)}$ as constants into the derivative and step (c) uses our induction hypothesis on $\vec{\tau}'$. Then, step (d) takes $\frac{1}{\mathrm{Z}}$ out of the derivative as we pushed it in as a constant. Finally, step (e) follows by the same reasoning as step (a) in the base case above. ■

A.3

Theorem 4. *Algorithm E_2 computes the second-order expectation of additively decomposable functions $r\colon \mathcal{T} \mapsto \mathbb{R}^R$ and $t\colon \mathcal{T} \mapsto \mathbb{R}^T$ in:*

$$\mathcal{O}(N^3 + N^2(\overline{R}\,\overline{T} + AR'T')) \text{ time}$$
$$\mathcal{O}(N^2 + RT + N(R+T)) \text{ space}$$

where $\overline{R} = \min(NR', R)$ and $\overline{T} = \min(NT', T)$.
Proof. We provide a proof of correctness (the time and space bounds are discussed in the main paper). Zmigrod et al. (2021) show that we can find second-order expectations over by finding the expectations over pairs of transitions. That is,

$$\mathbb{E}_{\tau_{i \rightsquigarrow \ell}} \left[r(\tau_{i \rightsquigarrow \ell}) t(\tau_{i \rightsquigarrow \ell})^\top \right] = \sum_{i,j,k,l=0}^{N} \sum_{a,b \in \overline{\mathcal{A}}} p\Big(i \xrightarrow{a} j, k \xrightarrow{b} l \Big) r_{ij}^{(a)} t_{kl}^{(b)^\top}$$

We can use Lemma 2 for the $m = 2$ case, to find that the expectation is given by

$$\begin{aligned}
&\mathbb{E}_{\tau_{i\rightsquigarrow\ell}}\Big[r(\tau_{i\rightsquigarrow\ell})t(\tau_{i\rightsquigarrow\ell})^\top\Big] \\
&= \frac{1}{\mathrm{Z}}\Big[\sum_{i,j=0}^{N}\sum_{a\in\overline{\mathcal{A}}}\frac{\partial \mathrm{Z}}{\partial \mathrm{W}_{ij}^{(a)}}\mathrm{W}_{ij}^{(a)}r_{ij}^{(a)}t_{ij}^{(a)\top} + \sum_{i,j,k,l=0}^{N}\sum_{a,b\in\overline{\mathcal{A}}}\frac{\partial^2 \mathrm{Z}}{\partial \mathrm{W}_{ij}^{(a)}\partial \mathrm{W}_{kl}^{(b)}}\mathrm{W}_{ij}^{(a)}\mathrm{W}_{kl}^{(b)}r_{ij}^{(a)}t_{kl}^{(b)\top}\Big]
\end{aligned}$$

The first summand can be rewritten as

$$\sum_{i,j=0}^{N}\sum_{a\in\overline{\mathcal{A}}}\frac{\partial \mathrm{Z}}{\partial \mathrm{W}_{ij}^{(a)}}\mathrm{W}_{ij}^{(a)}r_{ij}^{(a)}t_{ij}^{(a)\top} = \sum_{i,j=0}^{N}\sum_{a\in\overline{\mathcal{A}}}\mathrm{s}_i\dot{\mathrm{W}}_{ij}^{(a)}\mathrm{e}_j\mathrm{W}_{ij}^{(a)}r_{ij}^{(a)}t_{ij}^{(a)\top}$$

The second summand can be rewritten as

$$\begin{aligned}
&\sum_{i,j,k,l=0}^{N}\sum_{a,b\in\overline{\mathcal{A}}}\frac{\partial^2 \mathrm{Z}}{\partial \mathrm{W}_{ij}^{(a)}\partial \mathrm{W}_{kl}^{(b)}}\mathrm{W}_{ij}^{(a)}\mathrm{W}_{kl}^{(b)}r_{ij}^{(a)}t_{kl}^{(b)\top} \\
&\quad= \sum_{i,j,k,l=0}^{N}\sum_{a,b\in\overline{\mathcal{A}}}\mathrm{s}_i\dot{\mathrm{W}}_{ij}^{(a)}\mathrm{W}_{jk}^{\star}\dot{\mathrm{W}}_{kl}^{(b)}\mathrm{e}_l\mathrm{W}_{ij}^{(a)}\mathrm{W}_{kl}^{(b)}r_{ij}^{(a)}t_{kl}^{(b)\top} + \mathrm{s}_k\dot{\mathrm{W}}_{kl}^{(b)}\mathrm{W}_{li}^{\star}\dot{\mathrm{W}}_{ij}^{(a)}\mathrm{e}_j\mathrm{W}_{ij}^{(a)}\mathrm{W}_{kl}^{(b)}r_{ij}^{(a)}t_{kl}^{(b)\top}
\end{aligned}$$

Consider the first summand of the above expression

$$\begin{aligned}
&\sum_{i,j,k,l=0}^{N}\sum_{a,b\in\overline{\mathcal{A}}}\mathrm{s}_i\dot{\mathrm{W}}_{ij}^{(a)}\mathrm{W}_{jk}^{\star}\dot{\mathrm{W}}_{kl}^{(b)}\mathrm{e}_l\mathrm{W}_{ij}^{(a)}\mathrm{W}_{kl}^{(b)}r_{ij}^{(a)}t_{kl}^{(b)\top} \\
&=\sum_{j,k=0}^{N}\underbrace{\Big[\sum_{i=0}^{N}\sum_{a\in\overline{\mathcal{A}}}\mathrm{s}_i\dot{\mathrm{W}}_{ij}^{(a)}\mathrm{W}_{ij}^{(a)}r_{ij}^{(a)}\Big]}_{\stackrel{\text{def}}{=}\widehat{r_j^s}}\mathrm{W}_{jk}^{\star}\underbrace{\Big[\sum_{l=0}^{N}\sum_{b\in\overline{\mathcal{A}}}\dot{\mathrm{W}}_{kl}^{(b)}\mathrm{e}_l\mathrm{W}_{kl}^{(b)}t_{kl}^{(b)}\Big]^\top}_{\stackrel{\text{def}}{=}\widehat{t_k^e}^\top} \\
&=\sum_{j,k=0}^{N}\widehat{r_j^s}\mathrm{W}_{jk}^{\star}\widehat{t_k^e}^\top
\end{aligned}$$

Similarly, the second summand can be written as

$$\sum_{j,k=0}^{N}\widehat{r_k^e}\mathrm{W}_{jk}^{\star}\widehat{t_j^s}^\top$$

Finally, recomposing all the pieces together,

$$\mathbb{E}_{\tau_{i\rightsquigarrow\ell}}\Big[r(\tau_{i\rightsquigarrow\ell})t(\tau_{i\rightsquigarrow\ell})^\top\Big] = \frac{1}{\mathrm{Z}}\Big[\sum_{i,j=0}^{N}\widehat{r_i^s}\mathrm{W}_{ij}^{\star}\widehat{t_j^e}^\top + \widehat{r_j^e}\mathrm{W}_{ij}^{\star}\widehat{t_i^s}^\top + \sum_{a\in\overline{\mathcal{A}}}\mathrm{s}_i\dot{\mathrm{W}}_{ij}^{(a)}\mathrm{e}_j\mathrm{W}_{ij}^{(a)}r_{ij}^{(a)}t_{ij}^{(a)\top}\Big]$$

■

Reinforcement Learning for Abstractive Question Summarization with Question-aware Semantic Rewards

Shweta Yadav[*], **Deepak Gupta**[*], **Asma Ben Abacha**, **Dina Demner-Fushman**
LHNCBC, U.S. National Library of Medicine, MD, USA
{shweta.shweta, deepak.gupta, asma.benabacha}@nih.gov
ddemner@mail.nih.gov

Abstract

The growth of online consumer health questions has led to the necessity for reliable and accurate question answering systems. A recent study showed that manual summarization of consumer health questions brings significant improvement in retrieving relevant answers. However, the automatic summarization of long questions is a challenging task due to the lack of training data and the complexity of the related subtasks, such as the question focus and type recognition. In this paper, we introduce a reinforcement learning-based framework for abstractive question summarization. We propose two novel rewards obtained from the downstream tasks of (i) question-type identification and (ii) question-focus recognition to regularize the question generation model. These rewards ensure the generation of semantically valid questions and encourage the inclusion of key medical entities/foci in the question summary. We evaluated our proposed method on two benchmark datasets and achieved higher performance over state-of-the-art models. The manual evaluation of the summaries reveals that the generated questions are more diverse and have fewer factual inconsistencies than the baseline summaries. The source code is available here: https://github.com/shwetanlp/CHQ-Summ.

1 Introduction

The growing trend in online web forums is to attract more and more consumers to use the Internet for their health information needs. An instinctive way for consumers to query for their health-related content is in the form of natural language questions. These questions are often excessively descriptive and contain more than required peripheral information. However, most of the textual content is not particularly relevant in answering the question (Kilicoglu et al., 2013). A recent study showed that manual summarization of consumer health questions (CHQ) has significant improvement (58%) in retrieving relevant answers (Ben Abacha and Demner-Fushman, 2019). However, three major limitations impede higher success in obtaining semantically and factually correct summaries: **(1)** the complexity of identifying the correct question type/intent, **(2)** the difficulty of identifying salient medical entities and focus/topic of the question, and **(3)** the lack of large-scale CHQ summarization datasets. To address these limitations, this work presents a new reinforcement learning based framework for abstractive question summarization. We also propose two novel question-aware semantic reward functions: Question-type Identification Reward (QTR) and Question-focus Recognition Reward (QFR). The QTR measures correctly identified question-type(s) of the summarized question. Similarly, QFR measures correctly recognized key medical concept(s) or focus/foci of the summary.

We use the reinforce-based policy gradient approach, which maximizes the non-differentiable QTR and QFR rewards by learning the optimal policy defined by the Transformer model parameters. Our experiments show that these two rewards can significantly improve the question summarization quality, separately or jointly, achieving the new state-of-the-art performance on the MEQSUM and MATINF benchmark datasets. The main contributions of this paper are as follows:

- We propose a novel approach towards question summarization by introducing two question-aware semantic rewards **(i)** *Question-type Identification Reward* and **(ii)** *Question-focus Recognition Reward*, to enforce the generation of semantically valid and factually correct question summaries.
- The proposed models achieve the state-of-the-art performance on two question summa-

[*]These authors contributed equally to this work.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 249–255
August 1–6, 2021. ©2021 Association for Computational Linguistics

rization datasets over competitive pre-trained Transformer models.

- A manual evaluation of the summarized questions reveals that they achieve higher abstraction levels and are more semantically and factually similar to human-generated summaries.

2 Related Work

In recent years, reinforcement learning (RL) based models have been explored for the abstractive summarization task. Paulus et al. (2017) introduced RL in neural summarization models by optimizing the ROUGE score as a reward that led to more readable and concise summaries. Subsequently, several studies (Chen and Bansal, 2018; Pasunuru and Bansal, 2018; Zhang and Bansal, 2019; Gupta et al., 2020; Zhang et al., 2019b) have proposed methods to optimize the model losses via RL that enables the model to generate the sentences with the higher ROUGE score. While these methods are primarily supervised, Laban et al. (2020) proposed an unsupervised method that accounts for fluency, brevity, and coverage in generated summaries using multiple RL-based rewards. The majority of these works are focused on document summarization with conventional non-semantics rewards (ROUGE, BLEU). In contrast, we focus on formulating the semantic rewards that bring a high-level semantic regularization. In particular, we investigate the question's main characteristics, i.e., question focus and type, to define the rewards.

Recently, Ben Abacha and Demner-Fushman (2019) defined the CHQ summarization task and introduced a new benchmark (MEQSUM) and a pointer-generator model. Ben Abacha et al. (2021) organized the MEDIQA-21 shared task challenge on CHQ, multi-document answers, and radiology report summarization. Most of the participating team (Yadav et al., 2021b; He et al., 2021; Sänger et al., 2021) utilized transfer learning, knowledge-based, and ensemble methods to solve the question summarization task. Yadav et al. (2021a) proposed question-aware transformer models for question summarization. Xu et al. (2020) automatically created a Chinese dataset (MATINF) for medical question answering, summarization, and classification tasks focusing on maternity and infant categories. Some of the other prominent works in the abstractive summarization of long and short documents include Cohan et al. (2018); Zhang et al. (2019a); MacAvaney et al. (2019); Sotudeh et al. (2020).

3 Proposed Method

Given a question, the goal of the task is to generate a summarized question that contains the salient information of the original question. We propose a RL-based question summarizer model over the Transformer (Vaswani et al., 2017) encoder-decoder architecture. We describe below the proposed reward functions.

3.1 Question-aware Semantic Rewards

(a) Question-type Identification Reward: Independent of the pre-training task, most language models use maximum likelihood estimation (MLE)-based training for fine-tuning the downstream tasks. MLE has two drawbacks: (1) "*exposure bias*" (Ranzato et al., 2016) when the model expects gold-standard data at each step during training but does not have such supervision when testing, and (2) "*representational collapse*" (Aghajanyan et al., 2021), is the degradation of generalizable representations of pre-trained models during the fine-tuning stage. To deal with the *exposure bias*, previous works used the ROUGE and BLEU rewards to train the generation models (Paulus et al., 2017; Ranzato et al., 2016). These evaluation metrics are based on n-grams matching and might fail to capture the semantics of the generated questions. We, therefore, propose a new question-type identification reward to capture the underlying question semantics.

We fine-tuned a $\text{BERT}_{\text{BASE}}$ network as a question-type identification model to provide question-type labels. Specifically, we use the `[CLS]` token representation ($\boldsymbol{h}_{[CLS]}$) from the final transformer layer of $\text{BERT}_{\text{BASE}}$ and add the feed-forward layers on top of the $\boldsymbol{h}_{[CLS]}$ to compute the final logits

$$l = \boldsymbol{W}(tanh(\boldsymbol{U}h_{[CLS]} + \boldsymbol{a})) + \boldsymbol{b}$$

Finally, the question types are predicted using the *sigmoid* activation function on each output neuron of logits l. The fine-tuned network is used to compute the reward $r_{QTR}(Q^p, Q^*)$ as F-Score of question-types between the generated question summary Q^p and the gold question summary Q^*.

(b) Question-focus Recognition Reward: A good question summary should contain the key information of the original question to avoid factual inconsistency. In the literature, ROUGE-based rewards have been explored to maximize the coverage of the generated summary, but it does not guarantee to preserve the key information in the

question summary. We introduce a novel reward function called question-focus recognition reward, which captures the degree to which the key information from the original question is present in the generated summary question. Similar to QTR, we fine-tuned the $\text{BERT}_{\text{BASE}}$ network for question-focus recognition to predict the focus/foci of the question. Specifically, given the representation matrix ($\boldsymbol{H} \in \mathcal{R}^{n \times d}$) of n tokens and d dimensional hidden state representation obtained from the final transformer layer of $\text{BERT}_{\text{BASE}}$, we performed the token level prediction using a linear layer of the feed-forward network. For each token representation (h_i), we compute the logits $l_i \in \mathcal{R}^{|C|}$, where ($|C|$) is the number of classes and predict the question focus as follows: $f_i = softmax(\boldsymbol{W}h_i + \boldsymbol{b})$. The fine-tuned network is used to compute the reward $r_{QFR}(Q^p, Q^*)$ as F-Score of question-focus between the generated question summary Q^p and the gold question summary Q^*.

3.2 Policy Gradient REINFORCE

We cast question summarization as an RL problem, where the "*agent*" (ProphetNet decoder) interacts with the "*environment*" (Question-type or focus prediction networks) to take "*actions*" (next word prediction) based on the learned "policy" p_θ defined by ProphetNet parameters (θ) and observe "*reward*" (QTR and QFR). We utilized ProphetNet (Qi et al., 2020) as the base model because it is specifically designed for sequence-to-sequence training and it has shown near state-of-the-art results on natural language generation task. We use the REINFORCE algorithm (Williams, 1992) to learn the optimal policy which maximizes the expected reward. Toward this, we minimize the loss function $\mathcal{L}_{RL} = -E_{Q^s \sim p_\theta}[r(Q^s, Q^*)]$, where Q^s is the question formed by sampling the words q_t^s from the model's output distribution, i.e. $p(q_t^s | q_1^s, q_2^s, \ldots, q_{t-1}^s, \mathcal{S})$. The derivative of $\mathcal{L}_{RL}$ is approximated using a single sample along with baseline estimator b:

$$\bigtriangledown_\theta \mathcal{L}_{RL} = -(r(Q^s, Q^*) - b) \bigtriangledown_\theta logp_\theta(Q^s) \quad (1)$$

The Self-critical Sequence Training (SCST) strategy (Rennie et al., 2017) is used to estimate the baseline reward by computing the reward with the question generated by the current model using the greedy decoding technique, i.e., $b = r(Q^g, Q^*)$. We compute the final reward as a weighted sum of QTR and QFR as follows:

$$r(Q^p, Q^*) = \gamma_{QTR} \times r_{QTR}(Q^p, Q^*) + \gamma_{QFR} \times r_{QFR}(Q^p, Q^*) \quad (2)$$

We train the network with the mixed loss as discussed in Paulus et al. (2017). The overall network loss is as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{RL} + (1 - \alpha)\mathcal{L}_{ML} \quad (3)$$

where, α is the scaling factor and $\mathcal{L}_{ML}$ is the negative log-likelihood loss and equivalent to $-\sum_{t=1}^{t=m} logp(q_t^* | q_1^*, q_2^*, \ldots, q_{t-1}^*, \mathcal{S})$, where $\mathcal{S}$ is the source question.

4 Experimental Results & Analysis

4.1 Datasets

We utilized two CHQ abstractive summarization datasets: MEQSUM and MATINF[1] to evaluate the proposed framework. The MEQSUM[2] training set consists of $5,155$ CHQ-summary pairs and the test set includes 500 pairs. We chose 100 samples from the training set as the validation dataset.

For fine-tuning the question-type identification and question-focus recognition models, we manually labeled the MEQSUM dataset with the question type: (*'Dosage', 'Drugs', 'Diagnosis', 'Treatments', 'Duration', 'Testing', 'Symptom', 'Usage', 'Information', 'Causes'*) and foci. We use the labeled data to train the question-type identification and question-focus recognition networks. For question-focus recognition, we follow the BIO notation and classify each token for the beginning of focus token (**B**), intermediate of focus token (**I**), and other token (**O**) classes. Since, the gold annotations for question-types and question-focus were not available for the MATINF dataset, we used the pre-trained network trained on the MEQSUM dataset to obtain the silver-standard question-types and question-focus information for MATINF[3]. The MATINF dataset has $5,000$ CHQ-summary pairs in the training set and 500 in the test set.

4.2 Experimental Setups

We use the pre-trained uncased version[4] of ProphetNet as the base encoder-decoder model. We use a beam search algorithm with beam size 4 to decode the summary sentence. We train all summarization models on the respective training dataset for 20 epochs. We set the maximum question and summary sentence length to 120 and 20, respectively.

[1] Since the dataset was in Chinese, we translated it to English using Google Translate.

[2] https://github.com/abachaa/MeQSum

[3] https://github.com/WHUIR/MATINF

[4] https://huggingface.co/microsoft/prophetnet-large-uncased

	Models	MEQSUM			MATINF*		
		R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
Baselines	Seq2Seq (Sutskever et al., 2014)	25.28	14.39	24.64	17.77	5.10	21.48
	Seq2Seq + Attention (Bahdanau et al., 2015)	28.11	17.24	27.82	19.45	6.45	23.77
	Pointer Generator (PG) (See et al., 2017)	32.41	19.37	36.53	23.31	7.01	26.61
	SOTA (Ben Abacha and Demner-Fushman, 2019)	44.16	27.64	42.78	–	–	–
	SOTA* (Ben Abacha and Demner-Fushman, 2019)	40.00	24.13	38.56	24.58	7.30	28.08
	Transformer (Vaswani et al., 2017)	25.84	13.66	29.12	22.25	5.89	26.06
	BertSumm (Liu and Lapata, 2019)	26.24	16.20	30.59	31.16	11.94	34.70
	$T5_{BASE}$ (Raffel et al., 2019)	38.92	21.29	40.56	39.66	21.24	41.52
	PEGASUS (Zhang et al., 2019a)	39.06	20.18	42.05	40.05	23.67	43.30
	$BART_{LARGE}$ (Lewis et al., 2019)	42.30	24.83	43.74	42.52	23.13	43.98
	MINILM (Wang et al., 2020)	43.13	26.03	46.39	35.60	18.08	38.70
	ProphetNet (Qi et al., 2020)	43.87	25.99	46.52	46.94	27.77	48.43
	ProphetNet + ROUGE-L	44.33	26.32	46.90	48.17	28.13	48.66
Joint Learning	ProphetNet + Q-type	44.40	26.63	47.05	47.19	28.02	48.70
	ProphetNet + Q-focus	44.62	26.61	47.28	47.14	28.06	48.64
	ProphetNet + Q-type + Q-focus	44.67	26.72	47.34	47.18	28.04	48.65
Proposed Approach	ProphetNet + QTR	44.60	26.69	47.38	47.51	28.40	48.94
	ProphetNet + QFR	45.36	27.33	47.96	47.53	28.29	49.11
	ProphetNet + QTR + QFR	45.52	27.54	**48.19**	47.73	28.54	**49.33**

Table 1: Comparison of the proposed models and various baselines. SOTA* denotes the method trained on the same data that we used. MATINF* denotes a translated English subset of the original Chinese MATINF dataset.

Summary Label	MEQSUM				MATINF			
	M1	**M2**	**M3**	**M4**	**M1**	**M2**	**M3**	**M4**
Semantics Preserved (PC/FC)	14/19.5	9.5/29	18/28	19.5/29	6/32.5	9.5/33	13.5/34	14/35
Factual Consistent (PC/FC)	11/25	7.5/35	9.5/36.5	10/38	5.5/35	7/36	7.5/41	9/42.5
Incorrect	23	11	12.5	11	10.5	11.5	11.5	10
Acceptable	18.5	10	12.5	12.5	15	10.5	8.5	9.5
Perfect	8.5	29	25	26.5	24.5	28	30	30.5

Table 2: Results of the manual evaluation of the summaries generated by ProphetNet (M1), M1+QTR (M2), M1+QFR (M3), and M1+QTR+QFR (M4). For *Semantic Preserved* and *Factual Consistent*, we report the partially correct (PC) and fully correct (FC) numbers.

We first fine-train the proposed network by minimizing only the maximum likelihood (ML) loss. Next, we initialize our proposed model with the fine-trained ML weights and train the network with the mixed-objective learning function (Eq. 3). We performed experiments on the validation dataset by varying the α, γ_{QTR} and γ_{QFR} in the range of $(0, 1)$. The scaling factor (α) value 0.95, was found to be optimal (in terms of Rouge-L) for both the datasets. The values of $\gamma_{QTR} = 0.4$ and $\gamma_{QFR} = 0.6$ were found to be optimal on the validation sets of both datasets. To update the model parameters, we used Adam (Kingma and Ba, 2015) optimization algorithm with the learning rate of $7e-5$ for ML training and $3e-7$ for RL training. We obtained the optimal hyper-parameters values based on the performance of the model on the validation sets of MEQSUM and MATINF in the respective experiments. We used a cosine annealing learning rate (Loshchilov and Hutter, 2017) decay schedule, where the learning rate decreases linearly from the initial learning set in the optimizer to 0. To avoid the gradient explosion issue, the gradient norm was clipped within 1. For all the baseline experiments, we followed the official source code of the approach and trained the model on our datasets. We implemented the approach of Ben Abacha and Demner-Fushman (2019) to evaluate the performance on both datasets. All experiments were performed on a single NVIDIA Tesla V100 GPU having GPU memory of 32GB. The average runtimes (each epoch) for the proposed approaches M_2, M_3 and M_4 were $2.7, 2.8$ and 4.5 hours, respectively. All the proposed models have 391.32 million parameters.

4.3 Results

We present the results of the proposed question-aware semantic rewards on the MEQSUM and MATINF datasets in Table-1. We evaluated the generated summaries using the ROUGE (Lin, 2004) metric[5]. The proposed model achieves new state-of-the-art performance on both datasets by

[5] https://pypi.org/project/py-rouge/

Original Question-I: who makes bromocriptine i am wondering what company makes the drug bromocriptine... i have on my pituitary gland ... i have to buy them...
Reference: who manufactures bromocriptine?
Generated Summary
ProphetNet: what is bromocriptine? **Proposed Approach**: what company makes bromocriptine and how much does it cost?
Original Question-II: Have been on methadone for four years. I am interested in the rapid withdrawal under anesthesia, but do not have a clue where I can find a doctor or hospital who does this. I also would like to know the approximate cost and if or what insurance companies pay for this.
Reference: how can I find a physician (s) or hospital (s) who specialize in rapid methadone withdrawal under anesthesia, and the cost and insurance benefits for the procedure?
Generated Summary
ProphetNet: what is the treatment for rapid withdrawal of methadone under anesthesia? **Proposed Approach**: where can i find physician (s) who specialize in rapid withdrawal of methadone?

Table 3: Correct/Incorrect summaries generated on MEQSUM. Example-I shows a perfect summary over ProphetNet. The second example shows an incorrect summary with a partially extracted focus ('*under anesthesia*') and two missing types ('*cost*', '*procedures*').

outperforming competitive baseline Transformer models. We also compare the proposed model with the joint learning baselines, where we regularize the question summarizer with the additional loss obtained from the question-type (Q-type) identification and question-focus (Q-focus) recognition model. To make a fair comparison with the proposed approach, we train these joint learning-based models with the same weighted strategy shown in Eq. 3. The results reported in Table 1 show the improvement over the ProphetNet on both datasets.

In comparison to the benchmark model on MEQ-SUM, our proposed model obtained an improvement of 9.63%. A similar improvement is also observed on the MATINF dataset. Furthermore, the results show that individual QTR and QFR rewards also improve over ProphetNet and ROUGE-based rewards. These results support two major claims: **(1)** question-type reward assists the model to capture the underlying question semantics, and **(2)** awareness of salient entities learned from the question-focus reward enables the generation of fewer incorrect summaries that are unrelated to the question topic. The proposed rewards are model-independent and can be plugged into any pre-trained Seq2Seq model. On the downstream tasks of question-type identification and question-focus recognition, the pre-trained BERT model achieves the F-Score of 97.10% and 77.24%, respectively, on 10% of the manually labeled MEQ-SUM pairs.

Manual Evaluation: Two annotators, experts in medical informatics, performed an analysis of 50 summaries randomly selected from each test set. In MATINF, nine out of the 50 samples contained translation errors. We thus randomly replaced them. In both datasets, we annotated each summary with two labels '*Semantics Preserved*' and '*Factual Consistent*' to measure **(1)** whether the semantics (i.e., question intent) of the source question was preserved in the generated summary and **(2)** whether the key entities/foci were present in the generated summary. In the manual evaluation of the quality of the generated summaries, we categorize each summary into one of the following categories: '*Incorrect*', '*Acceptable*', and '*Perfect*'. We report the human evaluation results (average of two annotators) on both datasets in Table-2. The results show that our proposed rewards enhance the model by capturing the underlying semantics and facts, which led to higher proportions of perfect and acceptable summaries. The error analysis identified two major causes of errors: **(1)** Wrong question types (e.g. the original question contained multiple question types or has insufficient type-related training instances) and **(2)** Wrong/partial focus (e.g. the model fails to capture the key medical entities).

5 Conclusion

In this work, we present an RL-based framework by introducing novel question-aware semantic rewards to enhance the semantics and factual consistency of the summarized questions. The automatic and human evaluations demonstrated the efficiency of these rewards when integrated with a strong encoder-decoder based ProphetNet transformer model. The proposed methods achieve state-of-the-art results on two-question summarization benchmarks. In the future, we will explore other types of semantic rewards and efficient multi-rewards optimization algorithms for RL.

Acknowledgements

This research was supported by the Intramural Research Program of the National Library of Medicine, National Institutes of Health.

Ethics / Impact Statement

Our project involves publicly available datasets of consumer health questions. It does not involve any direct interaction with any individuals or their personally identifiable data and does not meet the Federal definition for human subjects research, specifically: "a systematic investigation designed to contribute to generalizable knowledge" and "research involving interaction with the individual or obtains personally identifiable private information about an individual."

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Asma Ben Abacha and Dina Demner-Fushman. 2019. On the role of question summarization and information source restriction in consumer health question answering. *AMIA Summits on Translational Science Proceedings*, 2019:117.

Asma Ben Abacha and Dina Demner-Fushman. 2019. On the summarization of consumer health questions. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2228–2234. Association for Computational Linguistics.

Asma Ben Abacha, Yassine Mrabet, Yuhao Zhang, Chaitanya Shivade, Curtis Langlotz, and Dina Demner-Fushman. 2021. Overview of the MEDIQA 2021 shared task on summarization in the medical domain. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 74–85, Online. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Deepak Gupta, Hardik Chauhan, Ravi Tej Akella, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Reinforced multi-task approach for multi-hop question generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2760–2775.

Yifan He, Mosha Chen, and Songfang Huang. 2021. damo_nlp at MEDIQA 2021: Knowledge-based preprocessing and coverage-oriented reranking for medical question summarization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 112–118, Online. Association for Computational Linguistics.

Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. 2013. Interpreting consumer health questions: The role of anaphora and ellipsis. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 54–62.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philippe Laban, Andrew Hsi, John Canny, and Marti A Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, volume 1.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Sean MacAvaney, Sajad Sotudeh, Arman Cohan, Nazli Goharian, Ish Talati, and Ross W Filice. 2019. Ontology-aware clinical abstractive summarization. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1013–1016.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Mario Sänger, Leon Weber, and Ulf Leser. 2021. WBI at MEDIQA 2021: Summarizing consumer health questions with generative transformers. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 86–95, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Sajad Sotudeh, Nazli Goharian, and Ross W Filice. 2020. Attend to medical ontologies: Content selection for clinical abstractive summarization. *arXiv preprint arXiv:2005.00163*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Canwen Xu, Jiaxin Pei, Hongtao Wu, Yiyu Liu, and Chenliang Li. 2020. Matinf: A jointly labeled large-scale dataset for classification, question answering and summarization. *arXiv preprint arXiv:2004.12302*.

Shweta Yadav, Deepak Gupta, Asma Ben Abacha, and Dina Demner-Fushman. 2021a. Question-aware transformer models for consumer health question summarization. *arXiv preprint arXiv:2106.00219*.

Shweta Yadav, Mourad Sarrouti, and Deepak Gupta. 2021b. NLM at MEDIQA 2021: Transfer learning-based approaches for consumer question and multi-answer summarization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 291–301, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D Manning, and Curtis P Langlotz. 2019b. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. *arXiv preprint arXiv:1911.02541*.

A Semantics-aware Transformer Model of Relation Linking for Knowledge Base Question Answering

Tahira Naseem, Srinivas Ravishankar, Nandana Mihindukulasooriya, Ibrahim Abdelaziz, Young-Suk Lee, Pavan Kapanipathi, Salim Roukos, Alfio Gliozzo, Alexander Gray
IBM Research
{tnaseem, ysuklee, kapanipa, roukos, gliozzo}@us.ibm.com
{srini, nandana.m, ibrahim.abdelaziz1, alexander.gray}@ibm.com

Abstract

Relation linking is a crucial component of Knowledge Base Question Answering systems. Existing systems use a wide variety of heuristics, or ensembles of multiple systems, heavily relying on the surface question text. However, the explicit semantic parse of the question is a rich source of relation information that is not taken advantage of. We propose a simple transformer-based neural model for relation linking that leverages the AMR semantic parse of a sentence. Our system significantly outperforms the state-of-the-art on 4 popular benchmark datasets. These are based on either DBpedia or Wikidata, demonstrating that our approach is effective across KGs.

1 Introduction

Knowledge base question answering (KBQA) has received significant interest due to its real-world applications. KBQA is a task where a natural language question is transformed into a precise structured query, using Entity Linking and Relation Linking as necessary sub-tasks to retrieve an answer. For example, the question *"Who founded the city where Pat Vincent died?"* requires mapping (a) *founded* and *died* to relations *dbo:founder* and *dbo:deathPlace*, and (b) entity *Pat Vincent* to *dbr:Pat_Vincent*, given DBpedia as the knowledge base.

Semantic parses such as Abstract Meaning Representation (AMR) have recently shown to be useful for the KBQA task (Lim et al., 2020). However, critical tasks for KBQA such as Relation Linking continue to be addressed primarily using the question text (Mulang' et al., 2020; Sakor et al., 2019b; Lin et al., 2020), ignoring the AMR parses of the question which can introduce additional semantics. In the literature, some systems such as SLING (Mihindukulasooriya et al., 2020) have used AMR for relation linking. However, similar to other rule-based approaches (Sakor et al., 2019b), SLING depends heavily on the specific target KG (DBpedia) and it is based on a complex ensemble of different approaches, making portability to new knowledge bases a non-trivial task.

In this work, we propose SemReL; a single Semantics-aware neural model for Relation linking. SemReL takes as input the question text annotated with its AMR parse and entity information and outputs a ranked list of relations. The key contributions of this work are as follows: (a) a simple, knowledge graph agnostic neural model for relation linking over knowledge bases, (b) leveraging AMR parses for better question representation, and (c) an experimental evaluation using four datasets based on DBpedia and Wikidata where we show that SemReL consistently outperforms existing systems on all datasets.

2 Semantics-aware Relation Linking

We propose a relation linking system that exploits the semantic structure of a sentence to retrieve relevant relations from the underlying knowledge base. We hypothesise that semantic representations abstract away from lexical forms, providing structural clues that are more consistent across training examples than surface text. To this end, we use the AMR graph of the sentence as its semantic structure. AMRs are directed acyclic graphs that capture *who is doing what to whom* in a sentence. The nodes in the graph are concepts and the edges are labelled with relations between those concepts. Figure 1 shows example AMR graphs for the question *"Who founded the city where Pat Vincent died?"*. Note that the AMR graph for the question represents the target of the query as a special node labelled 'amr-unknown'.

The inputs to our system are: the question text,

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 256–262
August 1–6, 2021. ©2021 Association for Computational Linguistics

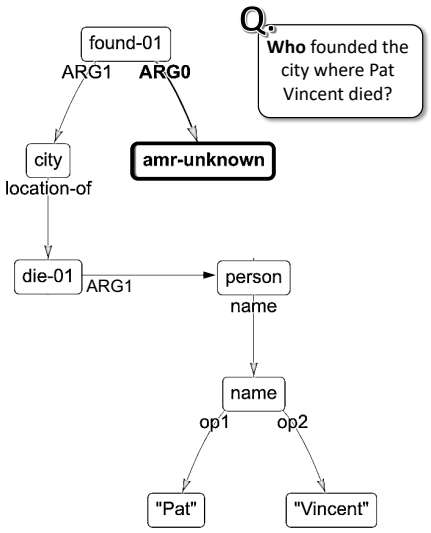


Figure 1: A question AMR: The *amr-unknown* represents the query target and the *name* node marks entities.

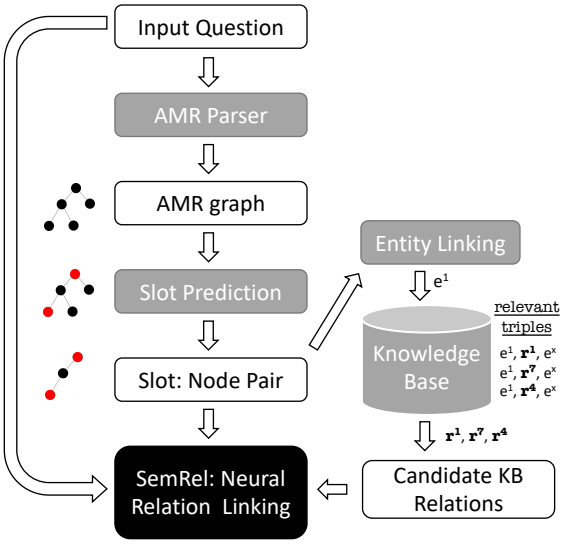


Figure 2: Overall system flow: grey blocks are various systems and white blocks show the inputs and outputs.

its AMR graph and the entities in the question marked and linked[1]. Relation linking is performed in two steps. First, our system identifies the number of expected relations and their location both in the sentence and in the AMR graph. Next, for each identified slot, the most likely relation is predicted using a transformer based neural model, that ranks them using their English labels from the KG. The AMR structure of the sentence is crucial in both steps. Figure 2 shows a schematic diagram of overall system. In the following, we first explain the process of finding potential relation slots using AMR graph. Next we describe in detail our relation linking module.

2.1 Relation Slot Prediction

AMR explicitly marks named entity nodes (see Figure 1). These nodes are linked to knowledge base entities using BLINK entity linker. The entity nodes in graph are also used to predict the number and locations of relation slots. A *slot* is defined as a pair of nodes in the AMR graph, where the corresponding entities have a relation in knowledge base in the context of the question. For instance, in Figure 3, nodes *city* and *person* are involved in a KB relation *death place* relevant for this question. Slot prediction is done using a deterministic rule-based transformation described in (Kapanipathi et al., 2021). In particular, we use their path-based approach where all the paths between the *amr-unknown* node and the linked entity nodes are retrieved. Then all node pairs along the path that are joined by a predicate node are considered a relation slot. We refer the reader to the original publication for more details of the method.

2.2 Neural Relation Linking Model

SemReL employs a Siamese network, where the input question and target relations are embedded in the same vector space. The most likely relation is the one whose representation is closest to that of the input question. Figure 3 shows the overall architecture of our model. We use a Transformer model (Vaswani et al., 2017) as a shared encoder for both the input questions and candidate relations. In particular, we use the pre-trained BERT model (Devlin et al., 2018) to initialize the encoder parameters. The output vector corresponding to the starting [CLS] token is used as the vector representation of the input. This vector is passed through a feed-forward linear layer that projects it to the shared embedding space. Unlike the transformer parameters, the weights of the linear projection layer on top are *not* shared between the questions and the relations.

Semantic information is given as part of the question input to the encoder. As mentioned above, during the preprocessing step, the pairs of nodes in AMR graph are identified for relation linking. For instance, in figure 3, the nodes 'person' and 'city' are marked in the input graph as the participants

[1]We use the stack transformer parser of Astudillo et al. (2020); Lee et al. (2020) for generating AMR graphs and the BLINK system of Wu et al. (2019) for entity linking.

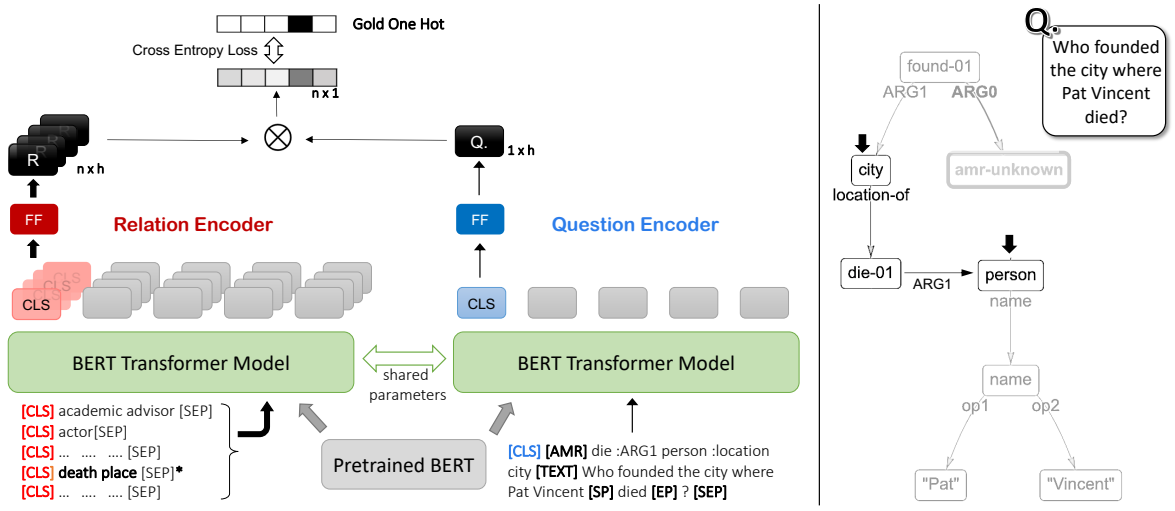


Figure 3: SemReL model architecture (left) and inputs to the model (right).

of a potential relation. The subgraph connecting these nodes is traversed in a top-down manner to form a linearized representation; in this case, it will yield the linearized string '*die :ARG1 person :location city*'. Note that the sense label of the node 'die-01' is dropped. Moreover, all reversed AMR relations with *-of* suffix are normalized to their original relation name and direction. In this example *:location-of* is mapped to *:location* with direction reversed. We prepend this linearized AMR path string to the input question text along with a special leading token [AMR]. The question text also starts with a special leading token [TEXT]. The word aligned to the root of the AMR subgraph is marked as the predicate[2], using special start and end predicate tokens [SP] and [EP].

Figure 3 shows the complete input for the example question that goes into the Question Encoder. The same transformer model also serves as relation encoder. Relation names are tokenized using BERT tokenizer without any additional pre-processing. We add special tokens [AMR], [TEXT], [SP] and [EP] as well as the AMR relation labels into the BERT vocabulary.

Training Objective: During training, for each example, scores are computed for the gold relation as well as a set of negative examples based on the inner product of their vectors with that of the question. For a relation r_i with vector representation $\mathbf{r_i}$ and a question q_n with vector representation $\mathbf{q_n}$, the score would be $s(r_i, q_n) = \mathbf{r_i}.\mathbf{q_n}$. The training objective is to minimize cross-entropy loss between the one-hot gold truth and the vector of predicted scores:

$$L(r_n, q_n) = -\log\left(\frac{\exp(s(r_n, q_n))}{\sum_i \exp(s(r_i, q_n))}\right)$$

We take the top one thousand relations from training data and use them as negative examples, excluding the gold. We compute the vector representation of all relations only once for each batch during training. Since relation representations are independent of the question representations, they can be reused for all examples in the batch. However, due to parameter update, they need to be computed anew for each batch.

Inference: During inference, we use $s(r, q)$ for scoring and ranking relations. Since the model parameters stay fixed, we compute the relation representations for all relations only once. If candidate KB relations are available from Entity analysis, we pick the highest-ranked relation from that set.

3 Evaluation

In this section, we detail our experimental setup and evaluate our approach against the state-of-the-art KBQA relation linking approaches. For fair comparisons, we replicate the same settings adopted by the systems we compare with both in terms of datasets and metrics.

3.1 Experimental Setup

Benchmarks: We perform experiments on four datasets targeting two popular KBs, DBpedia and

[2]The AMR parser of Astudillo et al. (2020) provides node to word alignments.

Wikidata. Each question in these datasets comes with its corresponding SPARQL query, annotated with gold relations. In particular, we used the following datasets:

- **QALD-9** (Usbeck et al., 2017): a dataset based on DBpedia with 150 test questions in natural language.
- **LC-QuAD 1.0** (Trivedi et al., 2017): another dataset based on DBpedia with a total of 5,000 questions (4,000 train and 1,000 test) based on templates.
- **LC-QuAD 2.0** (Dubey et al., 2019): A large dataset based on Wikidata with 6,046 test questions and around 24k training questions. Questions in this dataset have good variety and complexity levels such as multi-fact questions, temporal questions and questions that utilise qualifier information.
- **SimpleQuestions** (Diefenbach et al., 2017): A version of the popular SimpleQuestions dataset mapped to Wikidata. It comprises of 5,622 test questions, and around 19K training questions. As the name implies, all questions in this dataset are simple with queries encompassing a single triple in the KB.

Training: We train SemReL for DBpedia on the train data of LC-QuAD 1.0 and QALD-9. In addition, we use a subset of 80k examples from the distance supervisions data prepared by Mihindukulasooriya et al. (2020). This dataset is generated by retrieving Wikipedia sentences that contained pairs of entities from Knowledge Base triples. For our experiments, we filter our the sentences where the AMR path between the entities is more than two hops. For Wikidata experiments we train out system on the LC-QuAD 2.0 train dataset. Encoder parameters are initialized with the pretrained BERT base model (Wolf et al., 2020).

Baselines: For the DBpedia-based benchmarks, we compare SemReL with Falcon (Sakor et al., 2019a) and SLING (Mihindukulasooriya et al., 2020). As for Wikidata-based benchmarks, we compare against Falcon 2.0 (Sakor et al., 2020) and KB-Pearl (Lin et al., 2020).

[3]The KBPearl paper reports F1 of 0.41 due to a typo but its authors confirmed the correct F1 to be 0.52.

Dataset	Method	P	R	F1
QALD-9	Falcon	0.23	0.23	0.23
	SLING	0.39	0.50	0.44
	SemReL	0.46	0.44	**0.45**
LC-QuAD 1.0	Falcon	0.42	0.44	0.43
	SLING	0.41	0.55	0.47
	SemReL	0.51	0.51	**0.51**
LC-QuAD 2.0	Falcon 2.0	0.44	0.37	0.40
	SemReL	0.59	0.38	**0.46**
LC-QuAD 2.0 (1942 set)	KB-Pearl*	0.57	0.48	0.52[3]
	SemReL	0.70	0.45	**0.55**
Simple Questions	Falcon 2.0	0.35	0.44	0.39
	SemReL	0.69	0.70	**0.69**

Table 1: SemReL compared to SoTA systems on the DBpedia (above) and Wikidata (below) benchmarks.

Setup	all	one-hop	multi-hop
SemReL	0.51	0.54	0.50
w/o AMR	0.49	0.53	0.47
w/o TEXT	0.38	0.37	0.39
w/o KB rels	0.46	0.48	0.45

Table 2: SemReL F1 for *all*, *one-hop* and *multi-hop* questions with inputs ablated on LC-QuAD 1.0 testset. 'KB rels' refers to Knowledge Base relation candidates.

3.2 Results and Discussion

Table 1 compares SemReL with existing approaches. KB-Pearl used a subset of 1,942 test questions in their LC-QuAD 2.0 evaluation. For fair comparison, we also evaluate SemReL on the same subset.

SemReL outperforms all baselines across all benchmarks with respect to F1 score. Note that the baseline systems, specially SLING, achieve higher recall than precision. In contrast, SemReL has either balanced precision and recall, or much higher precision. This is in part due to missing entity or slot predictions, indicating that improving the preprocessing can further boost the system's performance. The results on SimpleQuestions are also worth noting, since the corresponding training set was not used in Wikidata model training. We also performed a zero-shot cross-KB experiment where we test our Wikidata model on a DBpedia dataset, LC-QuAD 1.0. The model is tested as is, and despite the relation names and granularity differences, it achieves an F1 of 0.33.

Ablation on Model Inputs: Table 2 shows the results of ablation experiments on LC-QuAD 1.0 testset where each of the system inputs are removed one at a time. As expected, the question text is the most crucial input: when combined with either KB candidates or AMR, it shows good performance. When AMR is removed, overall score drops by 2 points; it mostly comes from multi-hop questions. This indicates that focusing on different subgraphs of the input AMR improves retrieval of multi-hop relations. A similar effect was observed on QALD-9 and LC-QuAD 2.0 test sets when AMR was removed, degrading performance by 4.0 and 2.9 points respectively.

Impact on KBQA Performance: We integrated SemReL into the Neuro-Symbolic Question Answering (NSQA) system of Kapanipathi et al. (2021). NSQA is a modular system for KBQA where each sub-task is handled by a different module, allowing easy integration of new components. We found that the impact of using AMR in relation linking translates into nice performance gains in overall KBQA results. When AMR is incorporated in the relation linking module, the system performance on LC-QuAD 1.0 test dataset improves by 2.4 achieving a new state-of-the-art F1 of 44.5. We refer the reader to the NSQA paper (Kapanipathi et al., 2021) for more details on the system and experiments.

4 Related Work

Several relation linking systems have been proposed recently (Mulang et al., 2017; Singh et al., 2017; Dubey et al., 2018; Sakor et al., 2019a; Pan et al., 2019; Lin et al., 2020). Most of these methods are rule-based and rely solely on the question text and/or its dependency parse. Therefore, they try to improve their question understanding by using standard NLP tools such as POS tagging, tokenization n-gram tiling and even lexical database such as WordNet. FALCON 2.0 (Sakor et al., 2020) is a joint entity and relation linking tool over Wikidata. it uses a search engine indexed with Wikidata, a pipeline of text processing including POS tagging, tokenization, N-gram tiling/splitting and a catalog of rules for entity and relation linking. KBPearl (Lin et al., 2020) is another system that performs joint entity and relation linking to Wikidata. It first create a semantic graph of text using OpenIE and maps both entities and relations to a given KB.

However, none of the above mentioned methods for KBQA perform relation linking on two different KBs using the same system whilst our work is the first to perform relation linking over both DBpedia and Wikidata using the same system. In addition, some of these systems are KG-specific; e.g. Falcon (Sakor et al., 2019a) vs. Falcon 2.0 (Sakor et al., 2020), where adapting it from one KG to another requires non-trivial changes. Unlike these systems, SemReL leverages well-established semantic parsers such as AMR to achieve out-of-the-box better question representation.

Similar to our approach, SLING (Mihindukulasooriya et al., 2020) is a relation linking framework based on DBpedia which leverages semantic parsing using AMR and distant supervision. It consists of four distinct modules that capture different signals such as linguistic cues, semantic representation, and information from the knowledge base. Unlike SLING, SemReL is a KG-agnostic, single end-to-end neural model that does not require various ensemble components and yet achieves state-of-the-art performance on DBPedia and Wikidata datasets.

5 Conclusions and Future Work

In this paper, we present a simple transformer-based neural model for relation linking that leverages the semantic structure of a sentence. In contrast to existing systems such as SLING and Falcon, which are either rule-based or ensembles of several components, our neural architecture enables us to adapt the system to multiple KGs (e.g. DBpedia and Wikidata). It outperforms state-of-the-art systems on a variety of benchmarks.

Our ablation study shows that including the AMR graph improves performance, even with the relatively simple encoding scheme (plain text). In future, we will explore modeling the graph structure explicitly. This model also relies on a deterministic slot-finding algorithm based on AMR. While this identifies the correct relation slots most of the time, it is rule-based, and not always correct. In future work, we will explore learning algorithms to identify the slots from the AMR graph.

Finally, AMR parsers can be trained jointly with the relation linking objective. Currently, these parsers are sensitive to small changes in the input. Joint training can make them robust against text variations and more sensitive to the errors affecting slot identification and relation prediction.

References

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 1001–1007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dennis Diefenbach, Thomas Pellissier Tanon, Kamal Deep Singh, and Pierre Maret. 2017. Question answering benchmarks for wikidata. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017.*

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 69–78. Springer.

Mohnish Dubey, Debayan Banerjee, Debanjan Chaudhuri, and Jens Lehmann. 2018. Earl: joint entity and relation linking for question answering over knowledge graphs. In *ISWC*, pages 108–126.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramon Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, et al. 2021. Leveraging abstract meaning representation for knowledge base question answering. *Findings of the Association for Computational Linguistics: ACL 2021*.

Young-Suk Lee, Ramon Astuillo, Tahira Naseem, Revanth Reddy, Radu Florian, and Salim Roukos. 2020. Pushing the limits of amr parsing with self-learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 3208–3214.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. 2020. I know what you asked: Graph path learning using amr for commonsense reasoning. *arXiv preprint arXiv:2011.00766*.

Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. 2020. Kbpearl: a knowledge base population system supported by joint entity and relation linking. *Proceedings of the VLDB Endowment*, 13(7):1035–1049.

Nandana Mihindukulasooriya, Gaetano Rossiello, Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Mo Yu, Alfio Gliozzo, Salim Roukos, and Alexander Gray. 2020. Leveraging semantic parsing for relation linking over knowledge bases. In *International Semantic Web Conference*, pages 402–419. Springer.

Isaiah Onando Mulang', Jennifer D'Souza, and Sören Auer. 2020. Fine-tuning BERT with focus words for explanation regeneration. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Barcelona, Spain (Online). Association for Computational Linguistics.

Isaiah Onando Mulang, Kuldeep Singh, and Fabrizio Orlandi. 2017. Matching natural language relations to knowledge graph properties for question answering. In *SEMANTiCS 2017*, pages 89–96.

Jeff Z Pan, Mei Zhang, Kuldeep Singh, Frank van Harmelen, Jinguang Gu, and Zhi Zhang. 2019. Entity enabled relation linking. In *International Semantic Web Conference*, pages 523–538. Springer.

Ahmad Sakor, Isaiah Onando Mulang, Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2019a. Old is gold: linguistic driven approach for entity and relation linking of short text. In *NAACL: HLT 2019*, pages 2336–2346.

Ahmad Sakor, Isaiah Onando Mulang', Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2019b. Old is gold: Linguistic driven approach for entity and relation linking of short text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2336–2346, Minneapolis, Minnesota. Association for Computational Linguistics.

Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. Falcon 2.0: An entity and relation linking tool over wikidata. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3141–3148.

Kuldeep Singh, Isaiah Onando Mulang', Ioanna Lytra, Mohamad Yaser Jaradeh, Ahmad Sakor, Maria-Esther Vidal, Christoph Lange, and Sören Auer. 2017. Capturing knowledge in semantically-typed relational patterns to enhance relation linking. In *K-CAP 2017*, pages 1–8.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *ISWC 2017*, pages 210–218.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th open challenge on question answering over linked data (qald-7). In *Semantic Web Evaluation Challenge*, pages 59–69. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. *ArXiv*, abs/1911.03814.

Neural Retrieval for Question Answering with Cross-Attention Supervised Data Augmentation

Yinfei Yang[a], **Ning Jin**[b], **Kuo Lin**[b], **Mandy Guo**[a], **Daniel Cer**[a]

[a]Google Research
Mountain View, CA, USA

[b]Google Cloud AI
Sunnyvale, CA, USA

Abstract

Early fusion models with cross-attention have shown better-than-human performance on some question answer benchmarks, while it is a poor fit for retrieval since it prevents pre-computation of the answer representations. We present a supervised data mining method using an accurate early fusion model to improve the training of an efficient late fusion retrieval model. We first train an accurate classification model with cross-attention between questions and answers. The cross-attention model is then used to annotate additional passages in order to generate weighted training examples for a neural retrieval model. The resulting retrieval model with additional data significantly outperforms retrieval models directly trained with gold annotations on Precision at N (P@N) and Mean Reciprocal Rank (MRR).

1 Introduction

Open domain question answering (QA) involves finding answers to questions from an open corpus (Surdeanu et al., 2008; Yang et al., 2015; Chen et al., 2017; Ahmad et al., 2019). The task has led to a growing interest in scalable end-to-end retrieval systems for question answering.

When QA is formulated as a reading comprehension task, cross-attention models like BERT (Devlin et al., 2019) have achieved *better-than-human* performance on benchmarks such as the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016). Cross-attention models are especially well suited for problems involving comparisons between paired textual inputs, as they provide *early* fusion of fine-grained information within the pair. This encourages careful comparison and integration of details across and within the two texts.

However, early fusion across questions and answers is a poor fit for retrieval, since it prevents pre-computation of the answer representations. Rather, neural retrieval models independently compute embeddings for questions and answers, typically using dual encoders for fast scalable search (Henderson et al., 2017; Gillick et al., 2018; Yang et al., 2019b; Karpukhin et al., 2020). Using dual encoders results in *late* fusion within a shared embedding space.

For machine reading, early fusion using cross-attention introduces an inductive bias to compare fine grained text spans within questions and answers. This inductive bias is missing from the single dot-product scoring operation of dual encoder retrieval models. Thus, late fusion is expected to require more training data to learn the necessary representations for fine grained comparisons.

To support learning improved representations for retrieval, we explore a supervised data augmentation approach leveraging a complex classification model with cross-attention between question-answer pairs. Given gold question passage pairs, we first train a cross-attention classification model as the supervisor. Then any collection of questions can be used to mine potential question passage pairs under the supervision of the cross-attention model. The retrieval model training benefits from additional training pairs annotated with the graded predictions from the cross-attention model augmenting the existing gold data. Experiments on MultiReQA-SQuAD and MultiReQA-NQ establish significant improvements on Precision at N (P@N) and Mean Reciprocal Rank (MRR).

The supervised mining approach is closely connected to the recently studied hard negative mining for neural retrieval models (Xiong et al., 2020; Lu et al., 2020). The key differences is that the proposed approach finds the positive training examples, while the negative mining approaches find the negative examples for training. The two approaches are complementary and can be combined.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 263–268
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Neural Passage Retrieval for Open Domain Question Answering

Open domain question answering systems usually follow a two-step approach: first retrieve question relevant passages, and then scan the returned text to identify the answer span using a reading comprehension model (Jurafsky and Martin, 2018; Kratzwald and Feuerriegel, 2018; Yang et al., 2019a). Prior work has focused on the answer span annotation task and has even achieved super human performance on some datasets. However, the evaluations implicitly assume the trivial availability of passages for each question that are likely to contain the correct answer. While the retrieval task can be approached using traditional keyword based retrieval methods such as BM25, there is a growing interest in developing more sophisticated neural retrieval methods (Lee et al., 2019; Guu et al., 2020; Karpukhin et al., 2020).

3 Retrieval Question-Answering (ReQA)

Ahmad et al. (2019) introduced Retrieval Question-Answering (ReQA), a task that has been rapidly adopted by the community (Guo et al., 2020; Chang et al., 2020; Ma et al., 2020; Zhao and Lee, 2020; Roy et al., 2020). Given a question, the task is to retrieve the answer sentence from a corpus of candidates. ReQA provides direct evaluation of retrieval, independent of span annotation. Compare to Open Domain QA, ReQA focuses on evaluating the retrieval component and, by construction, avoids the need for span annotation.

We explore the proposed approach on MultiReQA-NQ and MultiReQA-SQuAD (Guo et al., 2020).[1] MultiReQA (Guo et al., 2020) established standardized training / dev / test splits. Statistics for each tasks are listed in Table 1.

Dataset	Training Pairs	Test	
		Questions	Candidates
NQ	106,521	4,131	22,118
SQuAD	87,133	10,485	10,642

Table 1: Statistics of MutiReQA NQ and SQuAD tasks: # of training pairs, # of questions, # of candidates.

4 Methodology

In this section we describe the proposed approach using a neural retrieval model augmented with supervised data mining. Figure 2 illustrates our approach using a cross-attention classifier to supervise the data augmentation process for training a retrieval model. After training the cross-attention model, we retrieve additional potential answers to questions using an off-the-shelf retrieval system[2]. The predicted scores from our classifier with cross-attention are then used to weight and filter the retrieved candidates with positive examples serving as additional training data for the dual encoder based retrieval model.

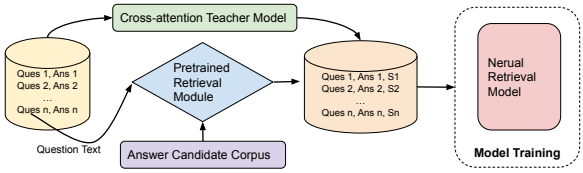


Figure 1: Use of a cross-attention model for the supervised mining of additional QA pairs. Our accurate cross-attention model supervises the mining process by identifying new previously unannotated positive pairs. Mined QA pairs augment the original training data for the dual encoder based neural passage retrieval model.

4.1 BERT Classification Model

Cross-attention models like BERT are often used for re-ranking after retrieval and can significantly improve performance as they allow for fine-grained interactions between paired inputs (Nogueira et al., 2019; Han et al., 2020). Here we formalize a binary classification task for predicting question answer relatedness. We use the question-answer pairs from the training set as our positive examples. Negatives are sampled for each question using the following strategies with a 1:1:1 ratio: *(1) A sentence from the top 10 nearest neighbors returned by a term based BM25 (Robertson and Zaragoza, 2009) over a sentence pool containing all supporting documents in a corpus. (2) A sentence from the top 10 nearest neighbors using the Universal Sentence Encoder - QA (USE-QA) (Yang et al., 2019b). (3) A sentence randomly sampled from its supporting documents, excluding the question's gold answer.* The sampled non-answer sentences are paired with their questions as negative examples. A BERT model is fine-tuned following the default setup from the Devlin et al. (2019).

[1] https://github.com/google-research-datasets/MultiReQA

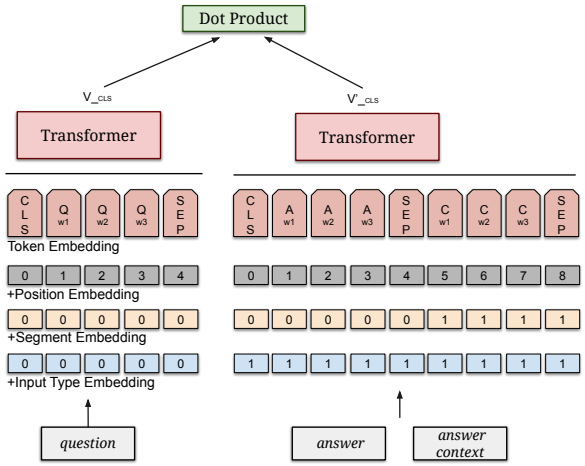


Figure 2: The BERT dual encoder architecture. The answer and context are concatenated and fed into the answer encoder. Figure from (Guo et al., 2020).

4.2 Dual-Encoder Retrieval Model

We follow Guo et al. (2020) and employ a BERT based dual-encoder model for retrieval. The model architecture is illustrated in figure **??**. The dual-encoder model critically differs from the cross-attention model in that there is no early interactions (cross-attention) between the question and answer. The resulting independent encodings are only combined in the final dot-product scoring a pair. The same BERT encoder is used for questions and answers with the output of the CLS token taken as the output encoding. For answers, the answer and context are concatenated and segmented using the segment IDs from the original BERT model. A learned *input type embedding* is added to each input token representation to distinguish questions and answers within the encoding model.

The BERT dual-encoder model can be fine-tuned using the in batch sampled softmax loss (Gillick et al., 2018):

$$\mathcal{J} = \sum_{(x,y)\in Batch} \frac{e^{\phi(x,y)}}{\sum_{\bar{y}\in\mathcal{Y}} e^{\phi(x,\bar{y})}} \quad (1)$$

Where x is the question, y is the correct answer, $\mathcal{Y}$ is all answers in the same batch that are used as sampled negatives, and $\phi(x, y)$ is the dot product of question and answer representations. Note that the dot product is scaled by X100 during training, which is a critical component when applying l_2 normalization to the embeddings.

[2]Note the approach can also be applied to any collection of questions, even for those without ground truth answers.

4.3 Mining Augmented Training Pairs

We create an augmented training set for the retrieval model using our cross-attention based QA model. For each question in the training set, we employ USE-QA to mine the top 10 nearest neighbors from the entire training set, and then remove those retrieved pairs which are true positives. Next the cross-attention based QA model is used to score the retrieved pairs. The dual-encoder based neural retrieval model is then trained on the combination the additional scored positive pairs and the original QA pairs from the training set. The original pairs are assigned a score 1.

4.4 Weighted In-batch Softmax for Dual-Encoder Retrieval Model

The neural retrieval model is trained using the batch negative sampling loss (Gillick et al., 2018) in equation 2. We modify the standard formulation to include a weight, $w(x, y)$, for each pair.

$$\mathcal{J}' = \sum_{(x,y)\in Batch} w(x,y) \frac{e^{\phi(x,y)}}{\sum_{\bar{y}\in\mathcal{Y}} e^{\phi(x,\bar{y})}} \quad (2)$$

We set $w(x, y)$ to 1 if (x, y) is a ground truth positive pair and $p(x, y)^2$, otherwise, whereby $p(x, y)$ is the probability from the cross-attention model.

5 Evaluation

In this section we evaluate the proposed approach using the MultiReQA evaluation splits for NQ and SQuAD. Models are assessed using Precision at N (P@N) and Mean Reciprocal Rank (MRR). Following the ReQA setup (Ahmad et al., 2019), we report P@N for N=$[1, 5, 10]$. P@N evaluates whether the true answer sentence appears in the top-N ranked candidates. MRR is calculated as MRR $= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$, where N is the total number of questions, and $rank_i$ is the rank of the first correct answer for the ith question.

5.1 Configurations

Our cross-attention QA models are fine-tuned from the public English BERT for 10 epochs, using a batch size of 256 and a weighted Adam optimizer with learning rate 3e-5. We experiment with both $\text{BERT}_{\text{Base}}$ and $\text{BERT}_{\text{Large}}$. All hyper-parameters are set using a dev set split out from the training data (10%). When mining for silver data, we only keep candidate pairs with positive cross-attention QA model scores (≥ 0.5).

Models	NQ		SQuAD	
	ACC	AUC-PR	ACC	AUC-PR
Majority	73.7	–	74.8	–
$\text{BERT}_{\text{dual\_encoder}}$	75.8	49.3	80.3	62.0
(x-attn) $\text{BERT}_{\text{Base}}$	84.3	92.8	92.6	96.5
(x-attn) $\text{BERT}_{\text{Large}}$	84.9	93.5	93.6	97.1

Table 2: Accuracy (ACC) and area under the precision-recall curve (AUC-PR) for the classification task. **Majority** is a simple baseline that always predicts false. **(x-attn)** indicates cross-attention QA models.

The $\text{BERT}_{\text{Base}}$ model is used to initialize the dual encoder retrieval model. During training we use a batch size of 64, and a weighted Adam optimizer with learning rate 1e-4. The maximum input length is set to 96 for questions and 384 for answers. Models are trained for 200 epochs. The embeddings are l_2 normalized. Hyper-parameters are manually tuned on a held out development set.

5.2 Performance for the Classification Task

The classification data created using the method from section 4.1 contains a total of 531k and 469k training examples for NQ and SQuAD, respectively. Test sets extracted from the SQuAD and NQ test splits contain 15k and 41k examples.[3]

Table 2 provides the performance of the cross-attention models, compared to a majority baseline which always predict false and a $\text{BERT}_{\text{dual\_encoder}}$ retrieval model without any mined examples that uses cosine similarity for prediction. Cross-attention based models outperform the baselines by a wide margin,[4] with $\text{BERT}_{\text{Large}}$ achieving the highest performance on all metrics. This is consistent with our hypothesis that early fusion models outperform late fusion based retrieval models. Both models achieve better performance on SQuAD than NQ. The SQuAD task has higher token overlap, as described in section 3, making the task somewhat easier. We use the $\textbf{BERT}_{\textbf{Large}}$ model to supervise the data augmentation in the next section.

5.3 Mined Examples

We mined the SQuAD and NQ training data to construct additional QA pairs. After collecting and scoring addition pairs using the method described in section 4.3, we obtained 53% (56,148) and 12% (10,198) more examples for NQ and SQuAD, respectively. Table 4 illustrated the examples retrieved by USE-QA and predicted as positive examples by our cross-attention QA classification model. Both examples are clear positive QA pairs.

Much less data is mined for SQuAD then NQ. We believe it is because of the way SQuAD was created, whereby workers write the questions based on the content of a particular article. The resulting questions are much more specific and biased toward a particular question types, e.g. *what* questions Ahmad et al. (2019). Additionally, the candidate pool for SQuAD is only half that of NQ, resulting in questions having fewer opportunities to be matched to good additional answers.

5.4 Results on the Retrieval QA

Table 3 gives P@N and MRR@100 for retrieval models on MultiReQA-SQuAD and MultiReQA-NQ. The first two rows show the result from two simple baselines: BM25 (Robertson and Zaragoza, 2009), USE-QA, and $\text{USE-QA}_{\text{finetune}}$ reported by Guo et al. (2020). BM25 remains a strong baseline, especially with 62.8% P@1 and 70.5% MRR for SQuAD. BM25's performance on NQ is much lower, as there is much less token overlap between NQ questions and answers. USE-QA matches the performance of BM25 on NQ but performs worse on SQuAD.[5] $\text{BERT}_{\text{dual\_encoder}}$ performs well compared to other baselines, especially on NQ with a +6.6 point improvement compared to the $\text{USE-QA}_{\text{finetune}}$ model.[6] Its P@1 on SQuAD performs better than USE-QA and BM25, but -3.1 points MRR worse than $\text{USQ-QA}_{\text{finetune}}$. On average, $\text{BERT}_{\text{dual\_encoder}}$ is the best among those baselines.

Performance improves by a large margin using augmented training data from our cross-attention QA model, obtaining a +8.6 and +7.0 improvement on NQ P@1 and MRR. Compare to NQ, the improvement on SQuAD is rather marginal. The augmented $\text{BERT}_{\text{dual\_encoder}}$ retrieval model only achieves slightly improved performance on SQuAD, with +1 points for both P@1 and MRR. As discussed in section 5.3, we mine much less data from SQuAD compare to NQ, with only 10% more data than the original training set. As demonstrated by the strong BM25 performance and shown in (Guo et al., 2020), the SQuAD QA pairs have high token overlap between question and answers,

[3] The positive / negative ratio is roughly 1:3.

[4] The poor performance of $\text{BERT}_{\text{dual\_encoder}}$ is also aligned with the hypothesis that cosine similarity score is not a globally consistent measurement of how good a pair (Guo et al., 2018).

[5] USE-QA can be fine-tuned, which usually significantly outperforms the default USE-QA model (Guo et al., 2020).

[6] Our $\text{Bert}_{\text{dual\_encoder}}$ performs better than the one reported in Guo et al. (2020), likely due to additional training epochs.

Models	NQ				SQuAD			
	P@1	P@5	P@10	MRR	P@1	P@5	P@10	MRR
BM25	24.7	–	–	36.6	62.8	–	–	70.5
USE-QA	24.7	–	–	34.7	51.0	–	–	62.1
USE-QA$_{\text{finetune}}$	38.0	–	–	52.3	**66.8**	–	–	**75.9**
BERT$_{\text{dual_encoder}}$	44.7	77.1	85.1	58.9	62.8	85.4	91.0	72.8
BERT$_{\text{dual_encoder}}$ Augmented	**53.3**	**82.3**	**88.5**	**65.9**	63.8	86.1	91.6	73.7

Table 3: Precision at N(P@N) (%) N=$[1, 5, 10]$ and Mean Reciprocal Rank (MRR) (%) on the MultiReQA tasks.

Score	Silver QA Pair
0.92	**Q:** what are the names of the two old muppets in the balcony that heckle everyone ? **A:** Statler and Waldorf are a pair of Muppet characters known for their cantankerous opinions and shared penchant for heckling.
0.90	**Q:** where the phrase dressed to the nines come from **A:** It appears in book six of Jean - Jacques Rousseau 's Confessions , his autobiography ...

Table 4: Mined positive examples identified using our cross-attention QA classification model.

minimizing the advantage of the neural methods in capturing more complex semantic relationships.

Effectiveness of Weighted Softmax. We further experimented the Retrieval QA tasks using the model with the non-modified softmax using the augmented data. All other configurations are keep the same. The MRR of the model using non-modified softmax is 60.1 on MultiReQA-NQ and 71.9 on MultiReQA-SQuAD, which are much worse than the model using weighted softmax. This result indicates the weighted softmax is important for the proposed approach.

6 Conclusion

In this paper, we propose a novel approach for making use of an early fusion classification model to improve late fusion retrieval models. The early fusion model is used for data mining to augment the training set for the late fusion model. The proposed approach mines 53% (56,148) and 12% (10,198) more examples for MultiRQA-NQ and MultiRQA-SQuAD, respectively. Compared to the models directly trained with gold annotations, the resulting retrieval models improve +8.6% and +1.0% P@1 on NQ and SQuAD respectively. The current pipeline assumes there exists annotated in-domain question answer pairs to train the cross-attention model. With a strong general purpose cross-attention model, our method could be modified to train in-domain retrieval models without gold data. We leave this to the future work.

References

Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. ReQA: An evaluation for end-to-end answer retrieval models. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 137–146, Hong Kong, China. Association for Computational Linguistics.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *CoRR*, abs/1811.08008.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Brussels, Belgium. Association for Computational Linguistics.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2020. MultiReQA: A cross-domain evaluation for retrieval question answering models. *arXiv preprint arXiv:2005.02507*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-rank with bert in tf-ranking. *arXiv preprint arXiv:2004.08476*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Daniel Jurafsky and James H. Martin. 2018. *Speech and Language Processing (3rd Edition, in draft)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Bernhard Kratzwald and Stefan Feuerriegel. 2018. Adaptive document retrieval for deep question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 576–581, Brussels, Belgium. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *ArXiv*, abs/1906.00300.

Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2020. Neural passage retrieval with improved negative contrast.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural retrieval via domain-targeted synthetic query generation.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. Lareqa: Language-agnostic answer retrieval from a multilingual pool.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of ACL-08: HLT*, pages 719–727, Columbus, Ohio. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval.

Wei Yang, Rui Qiao, Haocheng Qin, Amy Sun, Luchen Tan, Kun Xiong, and Ming Li. 2019a. End-to-end neural context reconstruction in Chinese dialogue. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 68–76, Florence, Italy. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019b. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

Tianchang Zhao and Kyusong Lee. 2020. Talk to papers: Bringing neural question answering to academic search. *arXiv preprint arXiv:2004.02002*.

Enhancing Descriptive Image Captioning with Natural Language Inference

Zhan Shi, Hui Liu, Xiaodan Zhu
Ingenuity Labs Research Institute, Queen's University
Department of Electrical and Computer Engineering, Queen's University
{z.shi, hui.liu, xiaodan.zhu}@queensu.ca

Abstract

Generating *descriptive* sentences that convey non-trivial, detailed, and salient information about images is an important goal of image captioning. In this paper we propose a novel approach to encourage captioning models to produce more detailed captions using natural language inference, based on the motivation that, among different captions of an image, descriptive captions are more likely to entail less descriptive captions. Specifically, we construct directed inference graphs for reference captions based on natural language inference. A PageRank algorithm is then employed to estimate the descriptiveness score of each node. Built on that, we use reference sampling and weighted designated rewards to guide captioning to generate descriptive captions. The results on MSCOCO show that the proposed method outperforms the baselines significantly on a wide range of conventional and descriptiveness-related evaluation metrics[1].

1 Introduction

Automatically generating visually grounded descriptions for given images, a problem known as image captioning (Chen et al., 2015), has drawn extensive attention recently. In spite of the significant improvement of image captioning performance (Lu et al., 2017; Anderson et al., 2018; Xu et al., 2015; Lu et al., 2018), existing models tend to *play safe* and generate generic captions. However, generating *descriptive* captions that carry detailed and salient information is an important goal of image captioning. For example, recent work (Luo et al., 2018; Liu et al., 2018b, 2019a) leveraged cross-modal retrieval (Faghri et al., 2017; Feng et al., 2014) to solve this problem, based on the observation that more *descriptive* captions often result in better discriminativity in retrieval.

In the paper, we explore to develop better descriptive image captioning models from a novel perspective— considering that among different captions of an image, descriptive captions are more likely to entail less descriptive ones, we develop descriptive image captioning models that leverage natural language inference (NLI, or also known as recognizing textual entailment) (Dagan et al., 2005; MacCartney and Manning, 2009; Bowman et al., 2015), which can utilize multiple references of captions (Young et al., 2014; Lin et al., 2014) to guide the models to produce more descriptive captions.

Specifically, the proposed model first predicts NLI relations for all pairs of references, i.e., *entailment* or *neutral*[2]. Built on that, we construct inference graphs and employ a PageRank algorithm to estimate descriptiveness scores for individual captions. We use reference sampling and weighted designated rewards to incorporate the descriptiveness signal into the Maximum Likelihood Estimation and Reinforcement Learning phase, respectively, to guide captioning models to produce *descriptive* captions. Extensive experiments were conducted on the MSCOCO dataset using different benchmark baseline methods (Huang et al., 2019; Luo et al., 2018; Rennie et al., 2017).

We demonstrate that the proposed method outperforms the baselines, achieving better performances on various evaluation metrics. In summary, the major contributions of the paper are three-fold: (1) To the best of our knowledge, this is the first attempt to connect natural language inference to image captioning, which helps generate more descriptive captions; (2) we propose a reference sampling distribution and weighted designated rewards to guide captioning model to produce more descriptive captions; (3) the proposed method attains better performance on various evaluation metrics over the

[1] https://github.com/Gitsamshi/Nli-image-caption

[2] As reference captions are unlikely to contradict to each other, we ignore the *contradiction* relation in our study.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 269–277
August 1–6, 2021. ©2021 Association for Computational Linguistics

state-of-the-art baselines.

2 Related Work

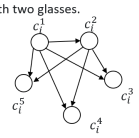
Image captioning Image captioning aims at generating visually grounded descriptions for images. It often leverages a CNN or variants as the image encoder and an RNN as the decoder to generate sentences (Vinyals et al., 2015; Karpathy and Fei-Fei, 2015; Donahue et al., 2015; Yang et al., 2016). To improve the performance on reference-based automatic evaluation metrics, previous work has used visual attention mechanism (Anderson et al., 2018; Lu et al., 2017; Pedersoli et al., 2017; Xu et al., 2015; Pan et al., 2020), explicit high-level attributes detection (Yao et al., 2017; Wu et al., 2016; You et al., 2016), reinforcement learning methods (Rennie et al., 2017; Ranzato et al., 2015; Liu et al., 2018a), contrastive or adversarial learning (Dai and Lin, 2017; Dai et al., 2017), multi-step decoding (Liu et al., 2019a; Gu et al., 2018), weighted training by word-image correlation (Ding et al., 2019) and scene graph detection (Yao et al., 2018; Yang et al., 2019; Shi et al., 2020).

The work of (Luo et al., 2018; Liu et al., 2018b) is most related to ours, which uses retrieval loss as a rewarding signal to encourage descriptive captioning. Different from the above approaches, our method explicitly explore the different *descriptiveness* in references using NLI models and incorporate the information into the training objectives to guide the model to generate more informative sentences. We build our method on top of the existing methods to verify the effectiveness.

Applications of NLI There are basically three major application types for NLI, (1) Direct application of trained NLI models. Trained NLI models are directly used in Fact Extraction and Verification (Thorne et al., 2018) to decide whether a piece of evidence supports a claim (Nie et al., 2019) and generation of longer sentences as a discriminator (Holtzman et al., 2018) to prevent a text decoder from contradicting itself; (2) NLI as a research and evaluation task for new methods. It is widely used as a major evaluation when developing novel language model pretraining (Devlin et al., 2018; Peters et al., 2018; Liu et al., 2019c); (3) NLI as a pre-training task in transfer learning. Training neural network models on NLI corpora and then fine-tuning them on target tasks often yields substantial improvements in performance (Liu et al., 2019b; Phang et al., 2018).

c_i^1: A glass for wine sitting next to a bottle.
c_i^2 : A wine bottle and some wine glasses in the hay.
c_i^3 : Two wine glasses lie beside a bottle of wine in straw.
c_i^4 : Two wine glasses lie beside a wine bottle on straw.
c_i^5 : A bottle of wine on it's side with two glasses.

$$\begin{array}{c} \\ c_i^1 \\ c_i^2 \\ c_i^3 \\ c_i^4 \\ c_i^5 \end{array} \begin{array}{c} \begin{array}{ccccc} c_i^1 & c_i^2 & c_i^3 & c_i^4 & c_i^5 \end{array} \\ \begin{pmatrix} e & n & n & n & n \\ e & e & n & n & n \\ e & e & e & e & e \\ e & e & e & e & e \\ e & e & e & e & e \end{pmatrix} \end{array}$$



Label	Description	Example
Paraphrase	Two way entailment, X entails Y and vice versa	X: Two wine glasses lie beside a bottle of wine in straw Y: Two wine glasses lie beside a wine bottle on straw
Forward Entailment	One way entailment, X entails Y and Y is neutral to X	X: Two wine glasses lie beside a bottle of wine in straw Y: A wine bottle and some wine glasses in the hay
Reverse Entailment	One way entailment, Y entails X and X is neutral to Y	X: A wine bottle and some wine glasses in the hay Y: Two wine glasses lie beside a bottle of wine in straw
Mutual Neutral	Two way neutral, X is neutral to Y and vice versa	X: a tall house with a large clock mounted on its face Y: a building and outdoor seating for a café

Figure 1: A NLI matrix and inference graph.

3 Our Method

The goal of image captioning is to train conditional generation model $p_\theta(c \mid x)$ based on training instances $(x_i, C_i)_{i=1}^m$ in a training dataset and $C_i = \{c_i^1, \cdots, c_i^n\}$, where m is the number of training instances and n is the number of reference captions for an image.

The typical models leverage a two-phase learning process to estimate $p_\theta(c \mid x)$: the first uses MLE objective, which minimizes a cross-entropy loss with regard to the ground truth captions:

$$\mathcal{L}_{\text{ML}}(\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{n} \log p_\theta(c_i^j \mid x_i) \qquad (1)$$

RL is then used to optimize models by maximizing the expected reward for generating captions.

$$\mathcal{L}_{\text{RL}}(\theta) = -\sum_{i=1}^{m} E_{\hat{c}\sim p_\theta(c|x_i)}[r(\hat{c}, x_i)] \qquad (2)$$

where $r(\hat{c}, x_i)$ could be CIDEr reward (r_{cd}) (Rennie et al., 2017) or a combination of CIDEr (r_{cd}) and discriminative loss (l_{dis}) (Luo et al., 2018).

In this work, we enhance these two basic learning objectives by considering the descriptiveness of references $\{c_i^1, \cdots, c_i^n\}$.

3.1 Constructing Inference Graphs

NLI Matrix The SNLI corpus (Bowman et al., 2015) is widely used for training natural language inference models. To leverage the data for our task, we extract a subset of SNLI to fit our needs, e.g., removing *contradiction* sentence pairs (see Appendix B for details). Our NLI model is built upon BERT (Devlin et al., 2018), which achieves near state-of-the-art performance and is sufficient for our purpose. Given reference captions $C_i =$

$\{c_i^1, \cdots, c_i^n\}$ of an image, we obtain a NLI label for each ordered pair $\langle c_i^j, c_i^k \rangle$, forming a NLI relation matrix, as shown in Figure 1. Note that a NLI relation matrix is not necessary to be a symmetric matrix. For example, it is possible that $\langle c_i^j, c_i^k \rangle$ has an entailment relation (i.e., c_i^j entails c_i^k) and $\langle c_i^k, c_i^j \rangle$ is neutral, by the definition in NLI (Bowman et al., 2015).

Inference Graphs Built on the NLI matrix, we construct the inference graphs. For c_i^j and c_i^k, if the ordered pair $\langle c_i^j, c_i^k \rangle$ and $\langle c_i^k, c_i^j \rangle$ are both *entailment* in the NLI matrix, c_i^j and c_i^k are *paraphrases*. If $\langle c_i^j, c_i^k \rangle$ is entailment and $\langle c_i^k, c_i^j \rangle$ is neutral, then $\langle c_i^j, c_i^k \rangle$ is said to be a *forward entailment* (FwdEntail). On the contrary, if $\langle c_i^j, c_i^k \rangle$ is neural and $\langle c_i^k, c_i^j \rangle$ is entailment, then $\langle c_i^j, c_i^k \rangle$ is said to be a *reverse entailment* (RevEntail). If both directions are neutral, we call it mutual neutral (muNeutral).

To construct a directed inference graph, captions in a given image are added as vertexes. We add a directed edge from c_i^j to c_i^k if $\langle c_i^j, c_i^k \rangle$ is revEntail; i.e., the edge's head c_i^k is expected to be more descriptive than the tail c_i^j, and the edge points towards c_i^k. If $\langle c_i^j, c_i^k \rangle$ is fwdEntail, we add an edge from c_i^k to c_i^j. We do not add edges for paraphrase and muNeural pairs.

Descriptiveness Scorer PageRank (Page et al., 1999) is a link analysis model applied to collections of nodes with quotations or references. We perform PageRank on a inference graph to compute the *descriptiveness* score for each node/caption, which measures at which node a random walk is more likely to stop. Nodes with a higher score assigned by PageRank can be viewed as more *descriptive*. We then normalize the score to obtain distribution $q(c \mid x_i), c \in C_i$.

3.2 Descriptiveness Regularized Learning

Reference sampling (Rs) for MLE We can verify that $\mathcal{L}_{\text{ML}}$ in Equation (1) is equivalent to the KL divergence between a uniform target reference distribution $U(c \mid x_i)$ and model distribution $p_\theta(c \mid x_i)$:

$$\mathcal{L}_{\text{ML}}(\theta) = \sum_{i=1}^{m} \text{KL}(U(c \mid x_i) || p_\theta(c \mid x_i)) \quad (3)$$

Note that Equation (3) indicates that any c that belongs to reference set of C_i will be equally learned without considering their *descriptiveness*. To resolve the issue, for an image x_i, we use the probability distribution q obtained from graph nodes. We obtain an enhanced MLE loss $\mathcal{L}'_{\text{ML}}$, which is equivalent to minimizing the KL divergence between the target reference sampling distribution q and p_θ:

$$\mathcal{L}'_{\text{ML}}(\theta) = \sum_{i=1}^{m} \text{KL}(q(c \mid x_i) || p_\theta(c \mid x_i)) \quad (4)$$

Weighted reward (Wr) for RL We modify the reward function in RL to integrate the *descriptiveness* score to encourage more contribution from descriptive references in designated reward. Specifically, we change the CIDEr reward item r_{cd} in $r(\hat{c}, x_i)$ as shown in equation (2) by replacing $U(c \mid x_i)$ with $q(c \mid x_i)$:

$$r'_{cd}(\hat{c}, x_i) = \sum_{j=1}^{n} q(c_i^j \mid x_i) \cdot \text{CD}(\hat{c}, c_i^j) \quad (5)$$

where CD denotes the CIDEr similarity score.

4 Experiment

4.1 Setup

Dataset and Evaluation Metrics We perform experiments on the Karpathy split of the MSCOCO dataset (Lin et al., 2014; Karpathy and Fei-Fei, 2015). We employ a wide range of conventional image caption evaluation metrics, i.e., SPICE(SP) (Anderson et al., 2016), CIDEr(CD) (Vedantam et al., 2015), METEOR(ME) (Denkowski and Lavie, 2014), ROUGE-L(RG) (Lin, 2004), and BLEU (Papineni et al., 2002) to evaluate the generated captions. Following (Liu et al., 2019a), we also use the caption generated $\hat{c}$ to retrieve image x using a separately trained image-matching model (Lee et al., 2018). The retrieval evaluation is based on 1K images (Lee et al., 2018) from the Karpathy test set. Retrieval performances are measured by R@K ($K = 1, 5$), i.e., whether x is retrieved within the top K retrieved images. We also perform human evaluation on *descriptiveness, fluency*, and *fidelity*.

Implementation Details To make a fair comparison, we use the same experiment setup that the compared baselines used. See more implementation details for NLI model, retrieval model in evaluation, and descriptiveness score normalization in appendix B.

Compared Models We use AoANet, ATTN, and DISC(λ set to 1) as the baselines. ATTN (Rennie et al., 2017) is a LSTM based decoder with

visual attention mechanism. AoANet (Huang et al., 2019) adopts the attention on attention module. We also leverage the discriminativity enhanced model DISC (Luo et al., 2018) which is built upon ATTN.

4.2 Results and Analyses

Overall Performance Table 1 shows the overall performance of different models.

Results on conventional metrics. Our method consistently outperforms the baseline models on most conventional metrics, especially SPICE and CIDEr; e.g., the proposed model improves the AoANet baseline from 118.4 to 119.1 on CIDEr, 21.5 to 21.7 on SPICE in the MLE phase, and improves the ATTN baseline on CIDEr from 117.4 to 120.1, SPICE from 20.5 to 21.0 in the RL phase. As CIDEr is based on tf-idf weighting, it helps to differentiate methods that generate more image-specific details that are less commonly occur across the dataset. As our method is designed to encourage models to generate sentences with more objects, attributes, or relations, the effect was also suggested by the improvement on SPICE.

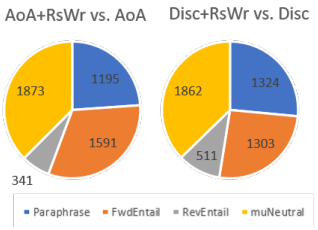


Figure 2: Inference labels in different models

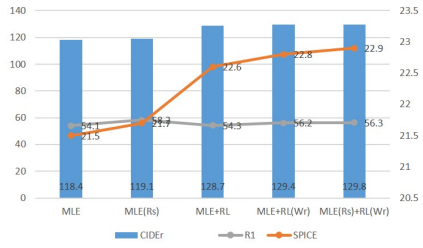


Figure 3: Ablation Analysis based on AoANet

Performance on descriptiveness related metrics. Our methods achieve consistently better results on R@1 and R@5 in both the MLE and RL optimization phases. Note that the proposed model can further boost the retrieval performance on the discriminativity enhanced baseline (DISC), improving R@1 from 46.5 to 48.1 and R@5 from 83.6 to 87.9. Our weighted CIDEr reward is complementary to the discriminative loss item in DISC and further boost the retrieval performance.

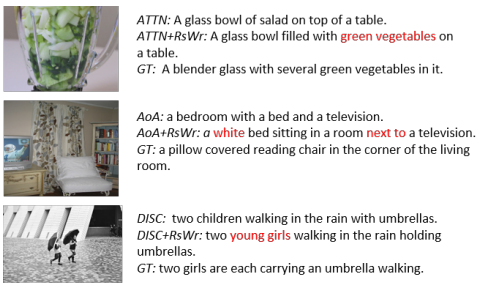


Figure 4: Examples generated by different models.

Labels between generated sentences. We use the externally trained NLI model (Section 3.1) to further investigate the NLI relationships between the captions generated by our method and by the baselines (AoA and DISC) on the testset. Figure 2 shows that our model generates more descriptive sentences. For example, comparing the generation results of AoA+RsWr and AoA on 5,000 testing images, captions generated by AoA+RsWr *forward-entails* those generated by AoA on 1,591 images, and *reverse-entails* on 341 images.

Ablation analysis. As shown in Figure 3, both reference sampling (Rs) and weighted reward (Wr) can improve performance in their respective optimization period, i.e., MLE to MLE(Rs), MLE+RL to MLE+RL(Wr). There is also a marginal improvement when using MLE(Rs) instead of MLE before the RL(Wr) optimization period, i.e., MLE+RL(Wr) to MLE(Rs)+RL(Wr), showing that MLE(Rs) has a positive impact even after RL(Wr) optimization.

Human Evaluation We further perform human evaluation on our method and two baselines (here, ATTN and DISC) using 100 images randomly sampled from the test set. Three human subjects rate captions with 1-5 Likert scales (higher is better) with respect to three criteria: *fluency*, *descriptiveness*, and *fidelity*. See more details in appendix A for rating details. Table 2 shows that ATTN+RsWr performs better than ATTN on descriptiveness. Moreover, DISC+RsWr can further improve the *descriptiveness* performance over the baseline discriminativity enhanced captioning model.

Case Study. Figure 4 includes three examples, in which our model produces captions with more attributes, objects, or relations.

5 Discussion

5.1 Descriptiveness and Entailment

We perform human analysis between descriptiveness and entailment. Specifically we randomly

	Maximum Likelihood Estimation							Reinforcement Learning						
	BLEU4	ME	RG	CD	SP	R@1	R@5	BLEU4	ME	RG	CD	SP	R@1	R@5
AoA	36.8	28.3	57.3	118.4	21.5	54.1	**87.6**	39.0	29.0	**58.9**	128.7	22.6	54.3	88.6
AoA+RsWr	**36.9**	**28.5**	**57.5**	**119.1**	**21.7**	**58.2**	87.4	39.0	**29.1**	58.7	**129.8**	**22.9**	**56.3**	**90.2**
ATTN	35.5	27.0	56.0	108.9	19.8	42.8	79.7	35.8	27.1	56.7	117.4	20.5	40.8	77.3
ATTN+RsWr	**35.8**	**27.3**	**56.3**	**112.1**	**20.5**	**48.2**	**84.4**	**36.2**	**27.3**	56.7	**120.1**	**21.0**	**44.9**	**84.8**
DISC	-	-	-	-	-	-	-	35.6	27.2	**57.0**	115.4	21.0	46.5	83.6
DISC+RsWr	-	-	-	-	-	-	-	**35.9**	27.2	56.8	**118.3**	**21.4**	**48.1**	**87.9**

Table 1: Results on MSCOCO karpathy split. RsWr detnotes Reference sampling and Weighted reward.

	Fluency	Descriptiveness	Fidelity
ATTN	3.90	2.53	3.46
ATTN+RsWr	**3.91**	**2.86**	**3.50**
DISC	**3.52**	3.08	3.28
DISC+RsWr	3.49	**3.30**	**3.31**

Table 2: Human evaluation on different models.

sample 50 images from the MSCOCO training set. For one image, there are five references, constituting ten reference pairs. So we have 500 reference pairs. For each reference pair, we ask three subjects to annotate whether one sentence conveys more non-trivial, important and detailed information than the other in terms of the described image. If the majority of the three subjects annotate yes, they further annotate the NLI relation—entailment or neutral, with the more informative caption as premise and the other as the hypothesis. As a result, out of the 500 reference pairs, we obtained 208 pairs that have differences in descriptiveness. The annotated NLI relations show that 164 of the 208 collected pairs have the entailment relation; i.e., for around 80% of the 208 pairs, "descriptive captions entail less descriptive captions" holds in the randomly sampled MSCOCO subset, where MSCOCO is a widely used multi-reference image caption benchmark.

5.2 Pairwise similarity and Re-ranking

We apply a pairwise similarity approach to AoA, in which we use Jaccard similarity between a pair of sentences to build the graph and run PageRank to get scores. Table 3 shows that pairwise similarity baseline approach (AoA+Sim) did not further improve performance over the corresponding baselines, showing pairwise similarity does not suggest descriptiveness, unlike entailment.

We perform re-ranking on the ATTN baseline; we use beam search with a beam size of 3, and then rank the captions in the beam by descriptiveness scores, which is calculated by BERT based NLI model. As shown in Table 3, the re-ranked sentences in the beam do not have much improvement in terms of baseline. Sentences generated by beam search (c.f. appendix C) do not vary significantly in terms of descriptiveness; these sentences are usually neutral to each other and sentences ranked low in the beam may have the fidelity/fluency issues.

	Pairwise Similarity Comparison						
	B@4	ME	RG	CD	SP	R@1	R@5
AoA	39.0	29.0	**58.9**	128.7	22.6	54.3	88.6
AoA+Sim	38.8	28.8	58.6	128.3	22.5	54.0	87.4
AoA+RsWr	39.0	**29.1**	58.7	**129.8**	**22.9**	**56.3**	**90.2**
	Re-ranking Comparison						
ATTN	35.8	27.1	56.7	117.4	20.5	40.8	77.3
ATTN+re-rank	35.7	27.2	**56.8**	117.0	20.6	41.5	78.8
ATTN+RsWr	**36.2**	**27.3**	56.7	**120.1**	**21.0**	**44.9**	**84.8**

Table 3: Comparison with pairwise similarity and re-ranking.

6 Conclusions

We explore a novel approach to encourage image captioning models to produce more descriptive sentences using natural language inference. We construct inference graphs and descriptiveness scores are assigned to nodes using the PageRank algorithm. Built on that, we use reference sampling and weighted designated rewards to guide captioning to generate descriptive captions. We demonstrate the effectiveness of the model on various evaluation metrics and perform detailed analyses.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. 2017. Towards diverse and natural image descriptions via a conditional gan. In *ICCV*.

Bo Dai and Dahua Lin. 2017. Contrastive learning for image captioning. In *Advances in Neural Information Processing Systems*, pages 898–907.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Guiguang Ding, Minghai Chen, Sicheng Zhao, Hui Chen, Jungong Han, and Qiang Liu. 2019. Neural image caption generation with weighted training and reference. *Cognitive Computation*, 11(6):763–777.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improved visual-semantic embeddings. *arXiv*, 2(7):8.

Fangxiang Feng, Xiaojie Wang, and Ruifan Li. 2014. Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 7–16.

Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. 2018. Stack-captioning: Coarse-to-fine learning for image captioning. In *AAAI*.

Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2018a. Context-aware visual policy network for sequence-level image captioning. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1416–1424.

Lixin Liu, Jiajun Tang, Xiaojun Wan, and Zongming Guo. 2019a. Generating diverse and descriptive image captions using visual paraphrases. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4240–4249.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. 2018b. Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In *ECCV*, pages 338–354.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural baby talk. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7219–7228.

Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. 2018. Discriminability objective for training descriptive captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6964–6974.

Bill MacCartney and Christopher D Manning. 2009. *Natural language inference*. Citeseer.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. 2020. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. 2017. Areas of attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1242–1250.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. 2020. Improving image captioning with better use of caption. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7454–7464.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton Van Den Hengel. 2016. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 203–212.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694.

Zhilin Yang, Ye Yuan, Yuexin Wu, William W Cohen, and Russ R Salakhutdinov. 2016. Review networks for caption generation. In *Advances in neural information processing systems*, pages 2361–2369.

Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699.

Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2017. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4894–4902.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *CVPR*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

A Human Evaluation Details

The human evaluation is performed with three non-author human subjects. We ask the subjects to rate on three 1-5 Likert scales, corresponding to *fidelity* (the sentences' fidelity to the corresponding images), *fluency* (the quality of captions in terms of grammatical correctness and fluency), and *descriptiveness* (how much the sentences convey more detailed and faithful information about the images).

B More Implementation Details

NLI We exclude the training instances labeled with *contradiction*, since our task does not need to consider contradiction—reference captions for the same image are unlikely to contradict each other. We also sample training instances in the SNLI dataset to make the subset's length distribution similar to the caption references. We obtained a filtered dataset with around 250K sentence pairs as our training set, 4K and 4K as validation and test set, respectively. We leverage BERT (Devlin et al., 2018) as the framework for training which is a basis for many state-of-the-art models and achieve near state-of-the-art performance, which is sufficient for our purpose. The training gets stabled after 3 epochs, reaching an accuracy around 88% on the test set.

Retrieval Model in Evaluation The model is trained with the published package of SCAN (Lee et al., 2018). For the specific parameters, we followed the "SCAN t-i LSE" setting in their published report.

Descriptiveness Score We use the entailment probability as the weights on the edges and then we perform PageRank using the toolkit from (Hagberg et al., 2008). We set the damping parameter of 0.95 for descriptiveness score at MLE training stage and 0.1 for descriptiveness score at RL training stage, as we find that a smooth score distribution on reward (c.f. Equation 5) and a peaked score distribution on MLE(c.f. Equation 4) lead to improved performance in the RL and MLE training stage respectively.

C Beam Search Generation

Example 1. {*"image˙id": 247625, "caption": a man holding a snowboard in the snow, a man standing on a snowboard in the snow, a man is standing on a snowboard in the snow*}

{*"image˙id": 131019, "caption": a group of zebras are standing in a field, a group of zebras are standing in a field with a zebra, a group of zebras are walking in a field*}

These are sentences generated by beam search by ATTN model after RL stage (before re-ranking).

MOLEMAN: Mention-Only Linking of Entities with a Mention Annotation Network

Nicholas FitzGerald, Jan A. Botha, Daniel Gillick, Daniel M. Bikel,
Tom Kwiatkowski, Andrew McCallum
Google Research
{nfitz,jabot,dgillick,dbikel,tomkwiat,mccallum}@google.com

Abstract

We present an instance-based nearest neighbor approach to entity linking. In contrast to most prior entity retrieval systems which represent each entity with a single vector, we build a contextualized mention-encoder that learns to place similar *mentions* of the same entity closer in vector space than mentions of different entities. This approach allows all mentions of an entity to serve as "class prototypes" as inference involves retrieving from the full set of labeled entity mentions in the training set and applying the nearest mention neighbor's entity label. Our model is trained on a large multilingual corpus of mention pairs derived from Wikipedia hyperlinks, and performs nearest neighbor inference on an index of 700 million mentions. It is simpler to train, gives more interpretable predictions, and outperforms all other systems on two multilingual entity linking benchmarks.

1 Introduction

A contemporary approach to entity linking represents each entity with a textual description d_e, encodes these descriptions and contextualized mentions of entities, m, into a shared vector space using dual-encoders $f(m)$ and $g(d_e)$, and scores each mention-entity pair as the inner-product between their encodings (Botha et al., 2020; Wu et al., 2019). By restricting the interaction between e and m to an inner-product, this approach permits the pre-computation of all $g(d_e)$ and fast retrieval of top scoring entities using maximum inner-product search (MIPS).

Here we begin with the observation that many entities appear in diverse contexts, which may not be easily captured in a single high-level description. For example, Actor Tommy Lee Jones played football in college, but this fact is not captured in the entity description derived from his Wikipedia page (see Figure 1). Furthermore, when new entities need to be added to the index in a zero-shot setting, it may be difficult to obtain a high quality description. We propose that both problems can be solved by allowing the entity mentions themselves to serve as exemplars. In addition, retrieving from the set of mentions can result in more interpretable predictions – since we are directly comparing two mentions – and allows us to leverage massively multilingual training data more easily, without forcing choices about which language(s) to use for the entity descriptions.

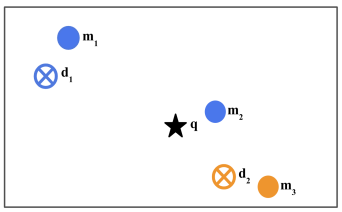
We present a new approach (MOLEMAN[1]) that maintains the dual-encoder architecture, but with the same mention-encoder on both sides. Entity linking is modeled entirely as a mapping between mentions, where inference involves a nearest neighbor search against all known mentions of all entities in the training set. We build MOLEMAN using exactly the same mention-encoder architecture and training data as Model F (Botha et al., 2020). We show that MOLEMAN significantly outperforms Model F on both the Mewsli-9 and Tsai and Roth (2016) datasets, particularly for low-coverage languages, and rarer entities.

We also observe that MOLEMAN achieves high accuracy with just a few mentions for each entity, suggesting that new entities can be added or existing entities can be modified simply by labeling a small number of new mentions. We expect this update mechanism to be significantly more flexible than writing or editing entity descriptions. Finally, we compare the massively multilingual MOLEMAN model to a much more expensive English-only dual-encoder architecture (Wu et al., 2019) on the well-studied TACKBP-2010 dataset (Ji et al., 2010) and show that MOLEMAN is competitive even in this setting.

[1]Mention Only Linking of Entities with a Mention Annotation Network

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 278–285
August 1–6, 2021. ©2021 Association for Computational Linguistics

Query Mention (q): "Harvard, with only five players of the 13, placed captain Vic Gatto and guard {Tom Jones} on the offensive team."



[Q170587]	$\mathbf{d}_1$	"Tommy Lee Jones is an American actor and film director."
	$\mathbf{m}_1$	"{Jones} moved to New York to become an actor in 1969 …"
	$\mathbf{m}_2$	"En Harvard {Jones} jugó como guard ofensivo en el equipo de fútbol americano universitario de 1968."
[Q18152778]	$\mathbf{d}_2$	"Thomas Lee Jones was an American and Canadian football player …"
	$\mathbf{m}_3$	"The Cleveland Browns selected {Tom Jones} …"

Figure 1: Illustration of hypothetical contextualized mention (m) and multilingual description (d) embeddings for the entities 'Tommy Lee Jones (Q170587)' and 'Tom Jones (Q18152778). The query mention [★] pertains to the former's college football career, which is unlikely to be captured by the high-level entity description. A retrieval against descriptions would get this query incorrect, but with indexed mentions gets it correct. Note that prior dual-encoder models that use a single vector to represent each entity are forced to contort the embedding space to solve this problem.

2 Overview

Task definition We train a model that performs entity linking by ranking a set of entity-linked *indexed mentions-in-context*. Formally, let a mention-in-context $\mathbf{x} = [x_1, ..., x_n]$ be a sequence of n tokens from vocabulary $\mathcal{V}$, which includes designated entity span tokens. An *entity-linked* mention-in-context $m^i = (\mathbf{x}^i, e^i)$ pairs a mention with an entity from a predetermined set of entities $\mathcal{E}$. Let $\mathcal{M}_\mathcal{I} = [m^1, ..., m^k]$ be a set of entity-linked mentions-in-context, and let $\texttt{entity}(\cdot) : \mathcal{M}_\mathcal{I} \rightarrow \mathcal{E}$ be a function that returns the entity $e^i \in \mathcal{E}$ associated with m^i, and $\mathbf{x}(\cdot)$ returns the token sequence $\mathbf{x}^i$.

Our goal is to learn a function $\phi(m)$ that maps an arbitrary mention-in-context token sequence m to a fixed vector $\mathbf{h}_m \in \mathcal{R}^d$ with the property that

$$y^* = \texttt{entity}\left(\underset{m' \in \mathcal{M}_I}{\text{argmax}}[\phi(\mathbf{x}(m'))^T \phi(\mathbf{x}_q)]\right)$$

gives a good prediction y^* of the true entity label of a query mention-in-context $\mathbf{x}_q$.

3 Method

3.1 Model

Recent state-of-the-art entity linking systems employ a dual encoder architecture, embedding mentions-in-context and entity representations in the same space. We also employ a dual encoder architecture but we score mentions-in-context (hereafter, mentions) against other mentions, with no consolidated entity representations. The dual encoder maps a pair of mentions (m, m') to a score:

$$s(m, m') = \frac{\phi(m)^T \phi(m')}{\|\phi(m)\| \|\phi(m')\|}$$

where ϕ is a learned neural network that encodes the input mention as a d-dimensional vector.

As in (Févry et al., 2020) and (Botha et al., 2020), our mention encoder is a 4-layer BERT-based Transformer network (Vaswani et al., 2017; Devlin et al., 2019) with output dimension $d = 300$.

3.2 Training Process

3.2.1 Mention Pairs Dataset

We build a dataset of mention pairs using the 104-language collection of Wikipedia mentions as constructed by Botha et al. (2020). This dataset maps Wikipedia hyperlinks to WikiData (Vrandečić and Krötzsch, 2014), a language-agnostic knowledge base. We create mention pairs from the set of all mentions that link to a given entity.

We use the same division of Wikipedia pages into train and test splits used by Botha et al. (2020) for compatibility to the TR2016 test set (Tsai and Roth, 2016). We take up to the first 100k mention pairs from a randomly ordered list of all pairs regardless of language, yielding 557M and 31M training and evaluation pairs, respectively. Of these, 69.7% of pairs involve two mentions from different languages. Our index set contains 651M mentions, covering 11.6M entities.

3.2.2 Hard Negative Mining and Positive Resampling

Previous work using a dual encoder trained with in-batch sampled softmax has improved performance with subsequent training rounds using an auxiliary cross-entropy loss against hard negatives sampled from the current model (Gillick et al., 2019; Wu et al., 2019; Botha et al., 2020). We investigate the effect of such negative mining for MOLEMAN, controlling the ratio of positives to negatives on a per-entity basis. This is achieved by limiting each entity to appear as a negative example at most 10

times as often as it does in positive examples, as done by Botha et al. (2020).

In addition, since MOLEMAN is intended to retrieve the *most similar* indexed mention of the correct entity, we experiment with using this retrieval step to resample the positive pairs used to construct our mention-pair dataset for the in-batch sampled softmax, pairing each mention m with the highest-scoring other mention m' of the same entity in the index set. This is similar to the index refreshing that is employed in other retrieval-based methods trained with in-batch softmax (Guu et al., 2020; Lewis et al., 2020a).

3.2.3 Input Representations

Following prior work (Wu et al., 2019; Botha et al., 2020), our mention representation consists of the page title and a window around the mention, with special mention boundary tokens marking the mention span. We use a total context size of 64 tokens.

Though our focus is on entity mentions, the entity descriptions can still be a useful additional source of data, and allow for zero-shot entity linking (when no mentions of an entity exist in our training set). We therefore experiment with adding the available entity descriptions as additional "pseudo-mentions". These are constructed in a similar way to the mention representations, except without mention boundaries. Organic and psuedo-mentions are fed into BERT using distinct sets of token type identifiers. We supplement our training set with additional mention pairs formed from each entity's description and a random mention, adding 38M training pairs, and add these descriptions to the index, expanding the entity set to 20M.

3.3 Inference

For inference, we perform a distributed brute-force maximum inner product search over the index of training mentions. During this search, we can either return only the top-scoring mention for each entity, which improves entity-based recall, or else all mentions, which allows us to experiment with k-Nearest Neighbors inference (see Section 4.1).

4 Experiments

4.1 Mewsli-9

Table 1 shows our results on the Mewsli-9 dataset compared to the models described by Botha et al. (2020). Model F is a dual encoder which scores entity mentions against entity descriptions, while Model F$^+$ adds two additional rounds of training with hard negative mining and an auxiliary cross-lingual objective. Despite using an identically-sized transformer, and trained on the same data, MOLEMAN outperforms Model F$^+$ when training only on mention pairs, and sees minimal improvement from a further round of training with hard negative and resampled positives (as described in Section 3.2.2). This suggests that training MOLEMAN is a simpler learning problem compared to previous models which must capture all an entity's diverse contexts with a single description embedding. Additionally, we examine a further benefit of indexing multiple mentions per entity: the ability to do top-K inference, and find that top-1 accuracy improves by half a point with k=5.

	I	HN	R@1	R@10	R@100
Model F	D	N	63.0	91.7	97.4
Model F$^+$	D	Y	89.4	96.4	98.2
MGENRE	–	–	90.6	–	–
MOLEMAN	M	N	89.5	97.4	98.3
	B	N	89.6	98.0	99.2
	B	Y	89.9	98.1	99.2
+ k=5	B	Y	90.4	–	–

Table 1: Results on Mewsli-9 compared to the models described by (Botha et al., 2020) and (De Cao et al., 2021). Column I indicates what is being indexed (Descriptions, Mentions, Both), and the HN indicates if additional rounds of Hard Negative training are applied.

We also compare to the recent MGENRE system of De Cao et al. (2021), which performs entity linking using constrained generation of entity names. It should be noted that this work uses an expanded training set that results in fewer zero- and few-shot entities (see De Cao et al. (2021) Table 3).

4.1.1 Per-Language Results

Table 2 shows per-language results for Mewsli-9. A key motivation of Botha et al. (2020) was to learn a massively multilingual entity linking system, with a shared context encoder and entity representations between 104 languages in the Wikipedia corpus. MOLEMAN takes a step further: the indexed mentions from all languages are included in the retrieval index, and can contribute to the prediction in any language. In fact, we find that for 21.4% of mentions in the Mewsli-9 corpus, MOLEMAN's top prediction came from a different language.

Language	R@1	R@10	R@100
ar	+1.1	+0.9	+0.3
de	-0.1	+1.5	+0.5
en	+0.3	+2.8	+2.3
es	-0.2	+1.1	+0.4
fa	+1.1	+0.9	+0.9
ja	+0.8	+1.2	+0.5
sr	-0.1	+0.8	+0.5
ta	+3.7	+1.3	+0.6
micro-avg	+0.2	+1.6	+1.0
macro-avg	+0.8	+1.3	+0.7

Table 2: MOLEMAN results on the Mewsli-9 dataset by language, listed as a delta against Model F^+ (Botha et al., 2020).

4.1.2 Frequency Breakdown

Table 3 shows a breakdown in performance by entity frequency bucket, defined as the number of times an entity was mentioned in the Wikipedia training set. When indexing only mentions, MOLEMAN can never predict the entities in the 0 bucket, but it shows significant improvement in the other frequency bands, particularly in the "few shot" bucket of [1,10). This suggests when introducing new entities to the index, labelling a small number of mentions may be more beneficial than producing a single description. To further confirm this intuition, we retrained MOLEMAN with a modified training set which had all entities in the [1, 10) band of Mewsli-9 removed, and only added to the index at inference time. This model achieved +0.2 R@1 and +5.6 R@10 relative to Model F^+ (which was trained with these entities in the train set). When entity descriptions are added to the index, MOLEMAN outperforms Model F^+ across frequency bands.

4.1.3 Inference Efficiency

Due to the large size of the mention index, nearest neighbor inference is performed using distributed maximum inner-product search. We also experiment with approximate search using ScaNN (Guo et al., 2020). Table 4 shows throughput and recall statistics for brute force search as well as two approximate search approaches that run on a single multi-threaded CPU, showing that inference over such a large index can be made extremely efficient with minimal loss in recall.

4.2 Tsai Roth 2016 Hard

In order to compare against previous multilingual entity linking models, we report results on the "hard" subset of Tsai and Roth (2016)'s cross-lingual dataset which links 12 languages to English Wikipedia. Table 5 shows our results on the same 4 languages reported by Botha et al. (2020). MOLEMAN outperforms all previous systems.

4.3 TACKBP 2010

Recent work on entity linking have employed dual-encoders primarily as a retrieval step before reranking with a more expensive cross-encoder (Wu et al., 2019; Agarwal and Bikel, 2020). Table 6 shows results on the extensively studied TACKBP 2010 dataset (Ji et al., 2010). Wu et al. (2019) used a 24-layer BERT-based dual-encoder which scores the 5.9 million entity descriptions from English Wikipedia, followed by a 24-layer cross-encoder reranker. MOLEMAN does not achieve the same level of top-1 accuracy as their full model, as it lacks the expensive cross-encoder reranking step, but despite using a single, much smaller Transformer and indexing the larger set of entities from multilingual Wikipedia, it outperforms this prior work in retrieval recall at 100.

We also report the accuracy of a MOLEMAN model trained only with English training data, and using an Enlish-only index for inference. This experiment shows that although the multilingual index contributes to MOLEMAN's overall performance, the pairwise training data is sufficient for high performance in a monolingual setting.

5 Discussion and Future Work

We have recast the entity linking problem as an application of a more generic mention encoding task. This approach is related to methods which perform clustering on test mentions in order to improve inference (Le and Titov, 2018; Angell et al., 2020), and can also be viewed as a form of cross-document coreference resolution (Rao et al., 2010; Shrimpton et al., 2015; Barhom et al., 2019). We also take inspiration from recent instance-based language modelling approaches (Khandelwal et al., 2020; Lewis et al., 2020b).

Our experiments demonstrate that taking an instance-based approach to entity-linking leads to better retrieval performance, particularly on rare entities, for which adding a small number of mentions leads to superior performance than a single description. For future work, we would like to explore the application of this instance-based approach to entity knowledge related tasks (Seo et al., 2018; Petroni et al., 2020), and to entity discovery (Ji et al., 2017).

	MOLEMAN (mentions only)		MOLEMAN (+ descriptions)		mGENRE
Freq. bin	R@1	R@10	R@1	R@10	R@1
[0, 1)	-8.3†	-33.9†	-0.2	+18.3	+13.8
[1, 10)	+0.4	+5.6	+1.7	+9.3	-10.4
[10, 100)	+1.9	+3.8	+1.7	+3.7	-3.1
[100, 1k)	+0.1	+1.8	-0.0	+1.9	+0.3
[1k, 10k)	-1.1	+0.7	-1.2	+0.7	+0.6
[10k,+)	+0.7	+0.6	+0.7	+0.5	+2.2
macro-avg	-1.1	-3.6	+0.5	+5.7	+0.6

Table 3: Results from MOLEMAN (with and without the inclusion of entity descriptions) on the Mewsli-9 dataset, by entity frequency in the training set plotted as a delta against Model F^+. †Note that when using mentions only, MOLEMAN scores zero on entities that do not appear in the training set.

	QPS	Latency (ms)	R@1	R@100
Brute-force	9.5	5727	89.9	99.2
ScaNN	8000	2.9	89.9	99.1

Table 4: Max throughput (queries per second), latency (ms per query) and recall for brute force inference and approximate MIPS inference using the ScaNN library (Guo et al., 2020). See Appendix A.3 for further details.

	MF+	MM
de	0.62	0.64
es	0.58	0.59
fr	0.54	0.58
it	0.56	0.59
Avg	0.57	0.60

Table 5: Accuracy results on the TR2016^{hard} test set for Model F^+ (MF+) and MOLEMAN (MM)

Method	**R@1**	**R@100**
AT-Prior	–	89.5
AT-Ext	–	91.7
BM25	–	68.9
Gillick et al. (2019)	–	96.3
Wu et al. (2019)	91.5†	98.3*
MOLEMAN (EN-only)	85.8	98.4
MOLEMAN	87.9	99.1

Table 6: Retrieval comparison on TACKBP-2010. The alias table and BM25 baselines are taken from Gillick et al. (2019). For comparison to Wu et al. (2019), we report R@1 for their "full Wiki, w/o finetune" cross-encoder. Their R@100 model is a dual-encoder fine-tuned on the TACKBP-2010 training set. MOLEMAN is not finetuned.

Acknowledgements

The authors would like to thank Ming-Wei Chang, Livio Baldini-Soares and the anonymous reviewers for their helpful feedback. We also thank Dave Dopson for his extensive help with profiling the brute-force and approximate search inference.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI 2016*.

Oshin Agarwal and Daniel M Bikel. 2020. Entity linking via dual and cross-attention encoders. *arXiv preprint arXiv:2004.03555*.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2020. Clustering-based inference for zero-shot biomedical entity linking. *arXiv preprint arXiv:2010.11253*.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *ACL 2019*.

Jan A Botha, Zifei Shan, and Dan Gillick. 2020. Entity linking in 100 languages. In *EMNLP 2020*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL 2019*.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. 2020. Empirical evaluation of pretraining strategies for supervised entity linking. In *AKBC 2020*.

Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *CoNLL 2019*.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *ICML 2020*.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *TAC 2010*.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, Cash Costello, and Sydney Informatics Hub. 2017. Overview of tac-kbp2017 13 languages entity discovery and linking. In *TAC 2017*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *ICLR 2020*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *ACL 2018*.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. In *NeurIPS 2020*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS 2020*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2020. KILT: a Benchmark for Knowledge Intensive Language Tasks.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *COLING 2010: Posters*.

Minjoon Seo, Tom Kwiatkowski, Ankur P Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. In *EMNLP 2018*.

Luke Shrimpton, Victor Lavrenko, and Miles Osborne. 2015. Sampling techniques for streaming cross document coreference resolution. In *NAACL 2015*.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *ACL 2016*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS 2017*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP 2020*.

A Appendices

A.1 Training setup and hyperparameters

To isolate the impact of representing entities with multiple mention embeddings, we follow the training methodology and hyperparameter choices presented in Botha et al. (2020) (Appendix A).

We train MOLEMAN using in-batch sampled softmax (Gillick et al., 2018) using a batch size of 8192 for 500k steps, which takes about a day. Our model is implemented in Tensorflow (Abadi et al., 2016), using the Adam optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) with the mention encoder preinitialized from a multilingual BERT checkpoint[2]. All model training was carried out on a Google TPU v3 architecture[3].

[2] github.com/google-research/bert/multi_cased_L-12_H-768_A-12

[3] cloud.google.com/tpu/docs/tpus

A.2 Datasets Links

- Mewsli-9: http://goo.gle/mewsli-dataset
- TR2016^{h}ard: cogcomp.seas.upenn.edu/page/resource_view/102
- TACKBP-2010: https://catalog.ldc.upenn.edu/LDC2018T16

A.3 Profiling Details

The brute-force numbers we've reported are the theoretical maximum throughput for computing 300D dot-products on an AVX-512 processor running at 2.2Ghz, and are thus an overly optimistic baseline. Practical implementations, such as the one in ScaNN, must also compute the top-k and rarely exceed 70% to 80% of this theoretical limit. The brute-force latency figure is the minimum time to stream the database from RAM using 144 GiB/s of memory-bandwidth. In practice, we ran distributed brute-force inference on a large cluster of CPUs, which took about 5 hours.

The numbers for ScaNN are empirical single-machine benchmarks of an internal solution that uses the open-source ScaNN library [4] on a single 24-core CPU. We use ScaNN to search a multi-level tree that has the following shape: $78,000 => 83:1 => 105:1$ (687.3 million datapoints). We used a combination of several different anisotropic vector quantizations that combine 3, 6, 12, or 24 dimensions per 4-bit code, as well as re-scoring with an `int8`-quantization.

A.4 Expanded experimental results

Tables 7 and 8 present complete numerical comparisons between MOLEMAN and Model F^{+} on Mewsli-9.

[4] https://github.com/google-research/google-research/tree/master/scann

	Model F+			MOLEMAN (mentions only)			MOLEMAN (+ descriptions)		
Language	R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@10	R@100
ar	92.3	97.7	99.1	93.4	98.6	99.0	93.4	98.6	99.4
de	91.5	97.3	99.0	91.3	98.2	98.9	91.5	98.9	99.5
en	87.2	94.2	96.7	87.4	95.9	97.4	87.4	97.0	99.3
es	89.0	97.4	98.9	88.7	98.1	98.8	88.7	98.5	99.3
fa	91.8	97.4	98.7	93.5	98.5	99.1	92.9	98.3	99.6
ja	87.8	95.6	97.6	88.7	96.2	97.0	88.5	96.8	98.0
sr	92.6	98.2	99.2	92.2	98.7	99.5	92.5	99.0	99.7
ta	87.6	97.4	98.9	91.5	98.4	99.1	91.3	98.6	99.5
micro-avg	89.4	96.4	98.2	89.5	97.4	98.3	89.6	98.0	99.2
macro-avg	89.8	96.9	98.5	90.6	97.8	98.5	90.6	98.2	99.3

Table 7: Results on the Mewsli-9 dataset by language.

		Model F+			MOLEMAN (mentions only)			MOLEMAN (+description)		
Bin	Queries	R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@10	R@100
[0, 1)	3,198	8.3	33.9	62.7	0.0	0.0	0.0	8.1	52.2	74.7
[1, 10)	6,564	57.7	80.8	91.3	58.1	86.4	93.3	59.4	90.1	96.5
[10, 100)	32,371	80.4	92.8	96.7	82.2	96.5	98.8	82.1	96.5	98.9
[100, 1k)	66,232	89.6	96.6	98.2	89.7	98.4	99.5	89.6	98.5	99.5
[1k, 10k)	78,519	92.9	98.4	99.3	91.9	99.2	99.8	91.8	99.1	99.8
[10k, +)	102,203	94.1	98.8	99.4	94.8	99.4	99.6	94.8	99.3	99.5
micro-avg		89.4	96.4	98.2	89.5	97.4	98.3	89.6	98.0	99.2
macro-avg		70.5	83.5	91.3	69.4	80.0	81.8	70.9	89.3	94.8

Table 8: Results on the Mewsli-9 dataset, by entity frequency in the test set.

eMLM: A New Pre-training Objective for Emotion Related Tasks

Tiberiu Sosea
Computer Science
University of Illinois at Chicago
tsosea2@uic.edu

Cornelia Caragea
Computer Science
University of Illinois at Chicago
cornelia@uic.edu

Abstract

Bidirectional Encoder Representations from Transformers (BERT) have been shown to be extremely effective on a wide variety of natural language processing tasks, including sentiment analysis and emotion detection. However, the proposed pre-training objectives of BERT do not induce any sentiment or emotion-specific biases into the model. In this paper, we present Emotion Masked Language Modeling, a variation of Masked Language Modeling, aimed at improving the BERT language representation model for emotion detection and sentiment analysis tasks. Using the same pre-training corpora as the original BERT model, Wikipedia and BookCorpus, our BERT variation manages to improve the downstream performance on 4 tasks for emotion detection and sentiment analysis by an average of 1.2% F1. Moreover, our approach shows an increased performance in our task-specific robustness tests. We make our code and pre-trained model available at https://github.com/tsosea2/eMLM.

1 Introduction

Language models have been studied extensively in the NLP community (Dai and Le, 2015; Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019), with approaches attaining state-of-the-art results on multiple token-level or sentence-level tasks. BERT (Devlin et al., 2019) is a pre-trained language model, which proposed a new pre-training objective inspired by the Cloze task (Taylor, 1953), which enables the training of a deep bi-directional transformer network. This objective, called Masked Language Modeling (MLM) is used on large amounts of unlabeled data from Wikipedia and BookCorpus to produce powerful universal language representations. However, the pre-training does not take into account the downstream task on which the model will be applied.

In this paper, we posit that we can leverage the characteristics of a downstream task to design better task-tailored pre-training objectives. Concretely, we induce information from emotion or sentiment lexicons into our BERT pre-training objective to improve the performance on tasks from sentiment analysis and emotion detection.

There are numerous studies that focus on emotion detection (Demszky et al., 2020; Desai et al., 2020; del Arco et al., 2020; Sosea and Caragea, 2020; Majumder et al., 2019; Mohammad and Kiritchenko, 2018; Abdul-Mageed and Ungar, 2017; Mohammad and Kiritchenko, 2015; Mohammad, 2012; Strapparava and Mihalcea, 2008) and sentiment analysis (Yin et al., 2020; Tian et al., 2020; Phan and Ogunbona, 2020; Zhai and Zhang, 2016; Chen et al., 2016; Liu, 2012; Glorot et al., 2011; Pang and Lee, 2005). Various lexicons have been used to improve model performance on these tasks. For instance, Katz et al. (2007) used occurrences of emotion words to identify various emotion types in news headlines. Moreover, emotion lexicons have been used to produce important features which can be used inside a machine learning algorithm to improve the performance on emotion detection tasks (Mohammad, 2012; Sykora et al., 2013; Khanpour and Caragea, 2018; Biyani et al., 2014). In this paper, however, instead of leveraging these lexicons to design features, in contrast, we use them to obtain language representations that are more suitable for emotion and sentiment tasks.

To this end, we introduce Emotion Masked Language Modeling (eMLM), a new pre-training BERT (Devlin et al., 2019) objective aimed at improving the BERT performance on tasks related to sentiment analysis and emotion detection. Inspired by the well-known Masked Language Modeling objective, eMLM adds only a few simple, yet powerful changes. Instead of uniformly masking the tokens in the input sequence, eMLM leverages

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 286–293
August 1–6, 2021. ©2021 Association for Computational Linguistics

SENT	They	look	absolutely	**perfect**	together	I	**hope**	its	that	way	in	real	life	too
MLM	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
eMLM	0.09	0.09	0.09	**0.50**	0.09	0.09	**0.50**	0.09	0.09	0.09	0.09	0.09	0.09	0.09
SENT	Most	**tiring**	thing	was	the	drive	one	hour	each	way				
MLM	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15				
eMLM	0.11	**0.50**	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11				

Table 1: Comparison of masked probabilities between MLM and eMLM on two example sentences.

lexicon information, and assigns higher masking probabilities to words that are more likely to be important in the sentiment or emotion contexts. To enable a fair comparison with the vanilla BERT model, we train the eMLM BERT model in the same fashion as the vanilla BERT, pre-training on Wikipedia and BookCorpus (Zhu et al., 2015). To our knowledge, we are the first to study different masking probabilities for the BERT pre-training procedure guided by sentiment and emotion lexicons. Similar to our work, some studies also focused on incorporating sentiment information into pre-trained language models. For example, Yin et al. (2020) built an attention network on top of BERT to predict sentiment labels of phrase nodes obtained through a constituency parse tree. On the other hand, Tian et al. (2020) designed various pre-training objectives, such as masking and predicting all words from a pre-defined small set of seeds, and predicting an aspect-sentiment pair or the polarity of words. In contrast, we leverage information from available sentiment and emotion lexicons.

We show the feasibility of our approach by testing eMLM on two sentiment analysis benchmark datasets and two emotion detection datasets. These datasets span diverse domains, such as movie reviews, online health communities, and Reddit discussions, enabling a comprehensive analysis of eMLM.

Our contributions are as follows: **1)** We introduce a new pre-training objective for BERT (leveraging available lexicons), aimed at producing better task-guided universal representations for downstream tasks from sentiment analysis and emotion detection. We offer the pre-trained model as an easy way to leverage our approach on downstream applications. **2)** We show the efficacy of our approach by testing our method on four benchmark datasets for emotion and sentiment and obtain an average improvement in F1 score of 1.2%. **3)** We verify the robustness of our model in the face of input perturbations, which occur frequently in informal contexts (e.g., due to mispellings).

2 Proposed Approach

Background Bidirectional Encoder from Transformers for Language Understanding (BERT) (Devlin et al., 2019) is a pre-trained language model trained on large amounts of unlabeled data using two objectives: **1)** Masked Language Modeling (MLM) randomly masks 15% of tokens in a sequence, followed by a supervised prediction of the masked tokens; **2)** Next Sentence Prediction (NSP) predicts in a binary fashion if two sentences follow each other. By using these two tasks on large-scale data repositories such as BookCorpus (800M words) (Zhu et al., 2015) and Wikipedia ($2,500$M words), BERT produces powerful universal language representations, applicable on a wide range of tasks, such as sentiment analysis, question answering, and commonsense reasoning.

However, to be used in various downstream tasks, BERT has to undergo a task-specific fine-tuning step (Devlin et al., 2019), where the contextualized embedding is adapted to the needed task. We posit that we can improve the downstream performance by focusing on the target task in the pre-training phase as well. Specifically, we focus on sentiment analysis and emotion detection, and show that task-guided unsupervised pre-training helps the performance considerably.

Masking Emotion Words Now we introduce Emotion Masked Language Modeling (eMLM), a variation of MLM targeted at inducing emotion or sentiment-specific biases in the BERT pre-training phase. Specifically, unlike BERT, which uses a uniform probability (15%) to mask the tokens in an input sentence, we assign higher probabilities to tokens which are emotionally rich words from an available lexicon $\mathcal{L}$. We denote this probability by k, which is a hyperparameter in our eMLM method. Our masking process can be summarized as follows: Given an input sentence S: **1)** We extract the words that belong to the lexicon $\mathcal{L}$, and we denote them by E; **2)** We set the masking probability of these words as $P(w_e) = k \; \forall \; w_e \in E$; **3)** To ensure

we mask 15% of the words in total, we lower the masking probability of the non-emotionally-rich words using the following formula:

$$P(w_n) = \frac{max(|S| \cdot 0.15 - |E| \cdot k,\ 0)}{|S| - |E|},\ \forall\ w_n \notin E$$

where $|\cdot|$ represents the size of a set. We show examples of how our masked probabilities change from MLM to eMLM in Table 1. For instance, in the first example, there are two emotion words, *perfect* and *hope*, and we use a masking probability of $k = 0.50$. While the probabilities of these two words are set to 50%, the non emotionally-rich word probability is lowered from 15% to 9% to keep the sum of probabilities constant. The rest of the training process is the same as the original BERT pre-training. That is, we train our BERT model from scratch using eMLM and NSP on the same datasets: Wikipedia and BookCorpus. We mention that we use whole word masking, both for eMLM and the MLM (i.e., we mask all the subtokens corresponding to a word).

3 Experiments and Results

In this section, we first describe our experimental setup (§3.1), then present our datasets and lexicons (§3.2), and then discuss the results that contrast **eMLM** with the original BERT **MLM** (§3.3).

3.1 Experimental Setup

We use various benchmark datasets from sentiment analysis and emotion detection to test our eMLM approach. For every dataset considered, we use the provided training, validation, and test splits. To assert statistical significance, we fine-tune each model 10 times with different random seeds and report the average F1 score. We investigate various masking probabilities k, ranging from 0.2 to 1.0, and find that 0.5 works best in our setting. For low values around 0.2 we notice that the performance is similar to that of the original BERT, while for high values (closer to 1.0), the performance is negatively affected.

3.2 Datasets and Lexicons

We test our models on various benchmark datasets described below.

Stanford Sentiment Treebank (**SST**) (Socher et al., 2013) SST contains 11,855 sentences from movie reviews, annotated with five sentiment labels: *negative*, *somewhat negative*, *neutral*, *somewhat positive*, and *positive*. First, we consider the binarized dataset, called SST-2, where the examples with the *negative* and *somewhat negative* labels are merged into a *negative* class, and the examples with the *somewhat positive* and *positive* labels are merged into a *positive* class (with neutral class being removed). Second, we consider the SST fine-grained version (SST-5), which uses all five labels.

	SST-2		SST-5	
	ACC	F-1	ACC	F-1
BERT	0.912	0.922	0.532	0.541
eMLM (S)	0.919	0.928	0.541	0.552
eMLM (E)	**0.920**	**0.931**†	**0.547**	**0.558**†

Table 2: Performance on the sentiment analysis task. We assert significance† if $p < 0.05$ under a t-test with the vanilla BERT model.

GoEmotions (Demszky et al., 2020) is a sentence-level multilabel dataset of 58,000 comments curated from Reddit and annotated with 27 emotion categories and the neutral class.

CancerEmo (Sosea and Caragea, 2020) is a sentence-level multilabel dataset of 8,500 sentences labeled with the eight Plutchik (Plutchik, 1980) basic emotions from an Online Health Community for people suffering from diseases such as cancer.

We analyze the behaviour of eMLM in diverse environments: sentiment analysis or emotion detection, various data platforms (e.g., Reddit, OHCs), and variate emotion or sentiment granularity (from 2 classes to as many as 28 classes).

Lexicons As mentioned above, our eMLM focuses on emotionally rich words from a lexicon. In this paper, we use EmoLex (Mohammad and Turney, 2013), a lexicon of 6,000 words associated with eight Plutchik basic emotions (Plutchik, 1980) (sadness, anger, joy, surprise, anticipation, trust, fear, disgust) and 5,555 words associated with the positive and negative sentiments. We consider the sentiment and emotion words separately to analyze the impact of each on the performance of eMLM. We denote the approach which masks the emotion-revealing words by eMLM (E), and the sentiment-revealing words by eMLM (S).

EMOTION	BERT	eMLM (E)	eMLM (S)
ADMIRATION	0.65	**0.68**†	0.67
AMUSEMENT	0.80	**0.83**†	0.82
ANGER	**0.47**	0.46	0.46
ANNOYANCE	0.34	0.34	0.34
APPROVAL	0.36	**0.38**	0.37
CARING	0.39	**0.43**	0.42
CONFUSION	0.37	0.37	0.37
CURIOSITY	0.54	**0.57**†	0.57
DESIRE	0.49	0.49	0.49
DISAPPOINTMENT	0.28	**0.30**	**0.30**
DISAPPROVAL	0.39	**0.43**†	0.41
DISGUST	0.45	**0.48**†	0.48
EMBARRASSMENT	0.43	0.43	**0.44**
EXCITEMENT	0.34	0.34	0.34
FEAR	0.60	**0.64**†	0.63
GRATITUDE	0.86	**0.88**†	0.87
GRIEF	0.00	0.00	0.00
JOY	0.51	**0.53**	0.52
LOVE	0.78	**0.80**†	**0.80**
NERVOUSNESS	0.35	**0.37**	0.36
NEUTRAL	**0.68**	0.67	**0.68**
OPTIMISM	0.51	**0.53**	0.52
PRIDE	0.36	0.36	0.36
REALIZATION	0.21	0.21	0.21
RELIEF	0.15	**0.16**	**0.16**
REMORSE	**0.66**	0.65	**0.66**
SADNESS	**0.49**	**0.49**	0.48
SURPRISE	0.50	**0.53**†	0.52
AVERAGE	0.462	**0.476**	0.469

Table 3: F-1 scores on the Goemotion dataset. We assert significance† if $p < 0.05$ under a t-test with the vanilla BERT model.

3.3 Results

Results on Sentiment Analysis We show the results of our approaches on SST in Table 2. First, we observe that eMLM (E) and eMLM (S) improve upon the vanilla BERT model on both tasks, with eMLM (E) obtaining as much as 1.7% improvement in F1. Interestingly, eMLM (E) outperforms eMLM (S) suggesting that masking finer-granularity emotion words in eMLM produces better representations for the task. At the same time, eMLM (E) achieves better performance on the fine-grained SST-5 task, where the improvements over the vanilla BERT are considerable.

Results on Emotion Detection We show the results of eMLM on the GoEmotions dataset in Table 3 and observe that, similar to sentiment analysis, eMLM (E) is the best performing approach, improving upon vanilla BERT by 1.4% in F1. We show the results on CancerEmo in Table 4 and observe the same pattern: **eMLM (E) consistently outperforms the other approaches**. We see improvements as high as 4% on Joy and 2% on Sadness. Overall, eMLM (E) obtains an 1.7% F1 improvement over the vanilla BERT model.

EMOTION	BERT	eMLM (E)	eMLM (S)
SADNESS	0.71	**0.73**†	**0.73**†
JOY	0.81	**0.85**†	0.84
FEAR	**0.77**	**0.77**	**0.77**
ANGER	0.68	**0.69**	**0.69**
SURPRISE	**0.68**	**0.68**	0.67
DISGUST	**0.59**	0.58	0.57
TRUST	0.67	0.67	0.67
ANTICIPATION	0.70	**0.78**†	0.74
AVERAGE	0.701	**0.718**	0.706

Table 4: Performance on CancerEmo dataset. We assert significance† if $p < 0.05$ under a t-test with the vanilla BERT model.

K	SST-2	SST-5	CANCEREMO	GOEMOTIONS
0.15	0.922	0.541	0.701	0.462
0.30	0.923	0.540	0.704	0.466
0.50	0.931	0.558	0.718	0.476
0.70	0.921	0.539	0.700	0.455
0.90	0.911	0.540	0.691	0.412

Table 5: Average F-1 on the considered datasets using various values of the emotion masking probability k.

Discussion The presented results reveal the feasibility of our proposed approach. Our BERT model trained using the eMLM objective produces high quality contextualized embeddings for downstream tasks that span the sentiment analysis and emotion detection tasks. Moreover, our methods incur no additional computational cost over the original BERT (Devlin et al., 2019), and undergo the same amount of pre-training. We also tried combining and masking both sentiment and emotion words; however, we did not see any performance improvements. As a step forward, we are interested in gaining more insights into the differences between eMLM (E) and the vanilla BERT model. We study this in the robustness context in the next section, and analyze how our models behave in the face of various input perturbations (i.e., noise).

Varying the Emotion Masking Probability k To offer additional insights into our eMLM approach and show the impact of the sentiment or emotion-rich word masking probability on downstream tasks, we show the results obtained using various values of k in Table 5. First, we note that using a slightly lower probability of 0.30 still adds improvements to our model on three of the considered datasets. In contrast, too high of a proba-

bility hurts the F1 performance. Concretely, using $k = 0.90$, our eMLM approach decreases the F1 compared to the vanilla BERT by 1% on **CancerEmo**, 5% on GoEmotions, and 1% on **SST-2**.

4 Robustness Test

It has been shown that neural models are often sensitive to various input perturbations (Niu et al., 2020; Belinkov and Bisk, 2018). In this section, we aim to investigate the robustness of our proposed approach in the face of input noise. We focus on the following two questions: **1)** Does eMLM improve the robustness of the model? **2)** What type of input noise is successful in misleading our model? We study these questions on the SST-5 sentiment analysis task using the framework introduced by Hsieh et al. (2019). We explore three ways to generate input perturbations and verify their "success." We say a perturbation is "successful" on a model M for an example e if **1)** The model M classifies e correctly and **2)** The model M misclassifies the example e when noise is applied to it. Naturally, the lower the perturbation success rate, the more robust a model is. The perturbations that we considered are as follows:

1. **Random** (Alzantot et al., 2018) replaces one word from the input sentence with a random word from the vocabulary. For a word, we repeat this process 100 times. If at least one of the replacements leads to an incorrect prediction, the perturbation is deemed to be successful.

2. **LIST** (Alzantot et al., 2018) replaces each word (one at a time) in the input text with a synonym. The input perturbation is successful if at least one replacement leads to an incorrect prediction.

3. **EmoWord** If there is an emotion word in the input sentence, then we zero out that word, otherwise, we zero out a random word from the input sequence.

Results We show the results of the robustness tests for the vanilla BERT and the eMLM approach in Table 6. First, EmoWord is the most successful perturbation, being twice as effective compared to the other methods. Second, we observe that Random and LIST obtain the same success rates among both the BERT and eMLM approach. However, on EmoWord, our eMLM approach is considerably more robust, outperforming the simple BERT model by 4.4%. We argue that this is the byproduct of the eMLM training procedure, which focuses on predicting emotion words in the pre-training step.

EMOTION	RANDOM	LIST	EMOWORD
BERT	1.5%	2.4%	9.8%
eMLM	1.5%	2.4%	**5.4%**

Table 6: Robustness of our models in terms of perturbation success rates. Lower success rates indicate more robust models.

5 Conclusion

In this paper, we introduced a new BERT pre-training objective suited for sentiment analysis and emotion detection tasks. We showed that the approach is feasible; it needs no additional pre-training compared to the vanilla BERT, and improves the performance by 1.2% F1 on average on various tasks. Our analysis also suggests that eMLM is more robust in the face of input perturbations. As future work, we note that our approach is general enough, so we plan to leverage different lexicons outside the sentiment analysis and emotion detection domains to investigate if the model generalizes well on other domains (e.g., financial). We also plan to study if our method is effective for non-English languages. Finally, we note that there exist lexicons that assign to words not only their emotion, but also their emotion intensity (Mohammad, 2018). Therefore, we plan to investigate if associating the masking probability with the emotion intensity (i.e., assign a higher probability to a more intensive word) would further help improve the performance.

Acknowledgments

We thank our anonymous reviewers for their constructive comments and feedback. This work is partially supported by the NSF Grants IIS-1912887 and IIS-1903963. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF. The computation for this project was performed on Amazon Web Services through a research grant.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. EmoNet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 718–728, Vancouver, Canada. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Flor Miriam Plaza del Arco, Carlo Strapparava, Luis Alfonso Ureña López, and María Teresa Martín Valdivia. 2020. Emoevent: A multilingual emotion corpus based on different events. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 1492–1498. European Language Resources Association.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Prakhar Biyani, Cornelia Caragea, Prasenjit Mitra, and John Yen. 2014. Identifying emotional and informational support in online health communities. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 827–836, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, Austin, Texas. Association for Computational Linguistics.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28, pages 3079–3087. Curran Associates, Inc.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Shrey Desai, Cornelia Caragea, and Junyi Jessy Li. 2020. Detecting perceived emotions in hurricane disasters. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5290–5305, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *In Proceedings of the Twenty-eight International Conference on Machine Learning, ICML*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529, Florence, Italy. Association for Computational Linguistics.

Phil Katz, Matt Singleton, and Richard Wicentowski. 2007. SWAT-MP:the SemEval-2007 systems for task 5 and task 14. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 308–313, Prague, Czech Republic. Association for Computational Linguistics.

Hamed Khanpour and Cornelia Caragea. 2018. Fine-grained emotion detection in health-related online posts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1160–1166, Brussels, Belgium. Association for Computational Linguistics.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander F. Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive RNN for emotion detection in conversations. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI*

2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 6818–6825. AAAI Press.

Saif Mohammad. 2012. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada. Association for Computational Linguistics.

Saif Mohammad. 2018. Word affect intensities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Saif Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Saif M. Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Minh Hieu Phan and Philip O. Ogunbona. 2020. Modelling context and syntactical features for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3220, Online. Association for Computational Linguistics.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Tiberiu Sosea and Cornelia Caragea. 2020. CancerEmo: A dataset for fine-grained emotion detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8892–8904, Online. Association for Computational Linguistics.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*, pages 1556–1560. ACM.

Martin D Sykora, Thomas Jackson, Ann O'Brien, and Suzanne Elayan. 2013. Emotive ontology: Extracting fine-grained emotions from terse, informal messages. *IADIS Int. J. Comput. Sci. Inf. Syst*, 2013:19–26.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4067–4076, Online. Association for Computational Linguistics.

Da Yin, Tao Meng, and Kai-Wei Chang. 2020. SentiBERT: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3695–3706, Online. Association for Computational Linguistics.

Shuangfei Zhai and Zhongfei (Mark) Zhang. 2016. Semisupervised autoencoder for sentiment analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1394–1400. AAAI Press.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies:

Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

On Positivity Bias in Negative Reviews

Madhusudhan Aithal
University of Colorado Boulder
madhuaithal@colorado.edu

Chenhao Tan
University of Chicago
chenhao@uchicago.edu

Abstract

Prior work has revealed that positive words occur more frequently than negative words in human expressions, which is typically attributed to positivity bias, a tendency for people to report positive views of reality. But what about the language used in negative reviews? Consistent with prior work, we show that English negative reviews tend to contain more positive words than negative words, using a variety of datasets. We reconcile this observation with prior findings on the pragmatics of negation, and show that negations are commonly associated with positive words in negative reviews. Furthermore, in negative reviews, the majority of sentences with positive words express negative opinions based on sentiment classifiers, indicating some form of negation.

1 Introduction

A battery of studies have validated the Pollyanna hypothesis that positive words occur more frequently than negative words in human expressions, using corpora ranging from Google Books to Twitter (Dodds et al., 2015; Garcia et al., 2012; Boucher and Osgood, 1969; Kloumann et al., 2012). The typical interpretation is connected with the positivity bias, which broadly denotes 1) a tendency for people to report positive views of reality, 2) a tendency to hold positive expectations, views, and memories, and 3) a tendency to favor positive information in reasoning (Carr, 2011; Augustine et al., 2011; Hoorens, 2014). However, it remains an open question whether the Pollyanna hypothesis holds in negative reviews, where the communicative goal is to express negative opinions.

In this work, we use a wide variety of review datasets to examine the use of positive and negative words in negative reviews. Table 1 shows a negative review from Yelp. Although the overall opinion is clearly negative, the author expressed the excitement to try the place and deemed the food OK. Zooming into individual words, they used the same number of positive and negative words in this negative review. Interestingly, this short review has as many as three negations, one directly applied to "great" (hence "not great").

Food was ok...*not* the money they charge. I was unimpressed and will *not* return. I was excited to try this place and was so disappointed as my expectations were high. Service *not* great and The parking is awful.

Table 1: Example negative review on Yelp. Positive words are in blue and negative words are in red, based on Vader (Hutto and Gilbert, 2014). Negations are in italics. This short review contains three negations.

More generally, we find that negative reviews contain *more* positive words than negative words, which is consistent with the Pollyanna hypothesis. Two possible reasons may explain this observation: 1) negative reviews tend to still include positive opinions due to a naïve interpretation of the positivity bias, where positive words express positive sentiments without accounting for negation or other contextual meaning of these words; 2) negative reviews tend to use *indirect expressions* (i.e., applying negations to positive words) to indicate negative opinions (e.g., "not clean"). Note that a broad interpretation of positivity bias may encompass the second reason,[1] but indirect expressions could also be related to other factors, e.g., verbal politeness (Brown et al., 1987)).

We aim to delineate these two reasons by examining the use of negations. Our results provide support for the latter reason: negative reviews tend to use more negations than positive reviews. The

[1] Boucher and Osgood (1969) used a morphological analysis to show negative affixes are more commonly applied to positive words than negative words (unhappy vs. non-violent).

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 294–304
August 1–6, 2021. ©2021 Association for Computational Linguistics

differences become even more salient when we compare negations applied to positive words vs. negative words. Finally, among sentences with positive words in negative reviews, the majority are classified as negative than as positive by sentiment classifiers, indicating some form of negation.

2 Related Work

In addition to positivity bias, our work is closely related to experimental studies on understanding the effect of direct (e.g., "bad") and indirect (e.g., "not good") wordings. Colston (1999) and Kamoen et al. (2015) observe no difference in people's interpretation of direct and indirect wordings in negative opinions; but direct wordings receive higher evaluations than indirect ones in positive opinions. In this work, we examine whether and how individuals use indirect wordings *in practice* (in negative reviews).

Our work is also related to Potts (2010), which finds that negation is used more frequently in negative reviews and is thus pragmatically negative. We extend Potts (2010) in two ways: 1) we demonstrate a high frequency of negation followed by positive words in negative reviews compared to other combinations, a new observation motivated through the lens of positivity bias; 2) we conduct a systematic study using a wide variety of datasets with multiple dictionaries.

Finally, our work builds on sentiment classification (Pang et al., 2002, 2008; Liu, 2012). The NLP community has made significant progress in recognizing the sentiment in texts of various languages, obtaining accuracies of over 95% (English) in binary classification (Devlin et al., 2019; Liu et al., 2019). Researchers have also developed novel approaches to identify fine-grained sentiments (e.g., aspect-level sentiment analysis (Schouten and Frasincar, 2015; Wang et al., 2016; Yang and Cardie, 2013)) as well as semi-supervised and unsupervised approaches (Hu et al., 2013; Zhou et al., 2010; Tan et al., 2011).

3 Datasets

We use a wide range of English review datasets to ensure that our results are robust across domains.

- Yelp.[2] We only consider restaurant reviews.
- IMDB movie reviews (Maas et al., 2011). This dataset provides train and test splits, so we follow their split when appropriate.

[2] https://www.yelp.com/dataset.

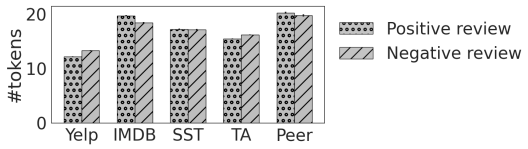


Figure 1: Sentence-length comparison. Although negative reviews can be much longer than positive reviews, sentences in positive reviews and negative reviews have similar lengths. Results on Amazon reviews are shown in the appendix. Tiny error bars show standard errors.

- Stanford sentiment treebank (SST) (Socher et al., 2013). SST contains processed snippets of reviews from the Rotten Tomatoes website (movie reviews). It has ground truth sentiment scores of reviews at the sentence level and the word level.
- Tripadvisor (Wang et al., 2010). This dataset consists of hotel reviews.
- PeerRead (Kang et al., 2018). We use reviews for papers in ACL, CoNLL, and ICLR.
- Amazon (Ni et al., 2019). This dataset contains Amazon reviews grouped by categories. We choose five categories that are substantially different from movies, hotels, and restaurants to ensure that our results are robust, namely, "Automotive", "Cellphones and accessories", "Luxury beauty", "Pet supplies", and "Sports and outdoors".

For datasets with ratings in 1-5 scale, we label reviews with ratings greater than 3 as positive and reviews with ratings less than 3 as negative following prior work (Pang et al., 2002), and ignore reviews with rating 3. Similarly, for datasets with ratings scale of 1-10 (IMDB, ICLR reviews in PeerRead), we label reviews with ratings greater than 6 as positive and review with ratings less than 5 as negative, and ignore reviews with ratings 5 and 6.

We use spaCy to tokenize the reviews in all datasets (Honnibal and Montani, 2017), except that Stanford Core NLP is used to tokenize SST reviews (Manning et al., 2014). We present results for Amazon reviews in the appendix, and our main results are robust on Amazon reviews. Our code is available at https://github.com/madhu-aithal/Positivity-Bias-in-Negative-Reviews.

Length of positive vs. negative opinions. In general, negative opinions tend to be longer than positive opinions ($p < 0.05$ after Bonferroni correction in 6 out 10 datasets; see the appendix for details). In comparison, the difference in length is smaller at the sentence level (Figure 1). Therefore, we use sentences as the basic unit in this work. To further

rule out sentence length as a confounding factor, we also present word-level results in the appendix.

4 Results

We first investigate the occurrences of positive words, negative words, and negations in reviews. We find that negative reviews contain more positive words than negative words in all datasets. We show that this observation relates to the prevalence of negation in negative reviews compared to positive reviews in all datasets. Furthermore, these negations are commonly associated with positive words in all datasets, and sentences with positive words tend to be negative based on sentence-level prediction, supporting the prevalence of indirect wordings in negative reviews.

4.1 Negative Reviews Have More Positive Words than Negative Words

We use lexicon-based methods to examine the frequency of positive and negative words in reviews. For most of the datasets, we randomly sample 5,000 positive reviews and 5,000 negative reviews to compute the lexicon distribution using LIWC (Pennebaker et al., 2007) and Vader (Hutto and Gilbert, 2014). In the case of SST, PeerRead, and negative reviews of Amazon Luxury Beauty, we use the entire dataset for our analysis as it has a relatively small number of reviews.

Figure 2 shows that as expected, negative reviews have more negative words and fewer positive words than positive reviews, based on Vader. Intriguingly, despite the negative nature of negative reviews, they tend to have more *positive* words than *negative* words ($p < 0.001$ on all datasets except SST after Bonferroni correction). Our results are robust at the word level and also hold based on LIWC and validate the Pollyanna hypothesis even in negative reviews.

4.2 Negative Reviews Have More Negations and Indirect Expressions

We hypothesize that in addition to the tendency to report positive views of reality, an important factor that can explain this observation in negative reviews is the use of indirect expressions (i.e., negation of positive words). To measure the amount of negation, we use two approaches: 1) a lexicon-driven approach based on Vader including *aint*, *cannot*, *not*, and *never* (Hutto and Gilbert, 2014)[3]; 2)

[3]See the appendix for the full list of negation lexicons.

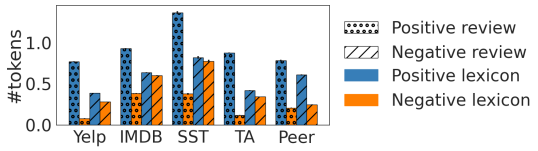
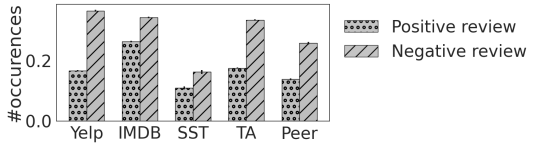
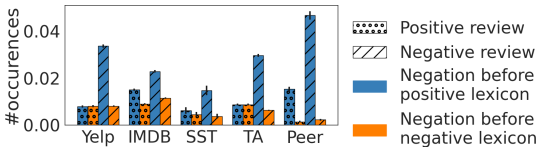


Figure 2: Number of positive and negative words based on Vader. Negative reviews have more positive words than negative words.



(a) Overall negation.



(b) Negation before positive and negative lexicons.

Figure 3: Negative reviews generally have more negations at the sentence level (Figure 3a). Among those negations, Figure 3b shows that there are substantially more negations before positive lexicons in negative reviews than any other combinations.

the negation relation in dependency parsing.[4] We present the results based on Vader negation in the main paper as it may have higher precision, and all results hold using dependency parsing.

Negative reviews have more negations than positive reviews in all datasets. Figure 3a presents the number of negations at the sentence level. In all datasets, negative reviews have more negations than positive reviews ($p < 0.001$ in all datasets). In fact, the number of negations in negative reviews almost doubles that in positive reviews in Yelp, TripAdvisor, and PeerRead (see samples in the appendix). This observation is robust at the word level, which accounts for the fact that negative reviews tend to be longer.

Negations are commonly associated with positive words in negative reviews. To further examine the relation between negations and sentimental lexicons, we investigate the occurrences of negations immediately followed by positive words and negative words. Figure 3b shows that there are more negations before positive words in negative reviews than any other combination ($p < 0.001$ in all datasets). The difference is especially salient in

[4]We used spaCy for dependency parsing (Honnibal and Montani, 2017).

Dataset	Positive words associated with negations
Yelp	recommend, sure, like, good, care, great, special, impressed, fresh, help, ready, enjoy, friendly, honor, helpful, clean, happy, accept, greeted, amazing
IMDB	like, care, funny, help, sure, recommend, good, save, fit, great, special, interesting, enjoy, well, play, better, giving, original, convincing, true
PeerRead	clear, sure, convincing, convinced, ready, well, true, clearly, surprising, novel, convincingly, recommend, guarantee, improve, interesting, support, satisfactory, help, acceptable, convince

Table 2: Most frequent positive words that immediately follow negations in negative reviews, based on Vader.

Yelp, TripAdvisor, and PeerRead. In particular, in negative reviews in PeerRead, negation before positive lexicon are approximately 20 times as frequent as negation before negative lexicon. These results demonstrate the prevalence of indirect wordings when people express negative opinions. Moreover, using indirect expressions to express negative opinions (negation before positive words) is also common in positive reviews for IMDB and PeerRead.

Table 2 shows the 20 most common words that immediately follow negations in Yelp, IMDB, and PeerRead, highlighting the prevalence of "not clear", "not convincing", and "not surprising" in negative reviews of NLP/ML submissions.

A natural question is how much of the usage of positive words in Figure 2 can be explained by negations before positive words. We find that it is sufficient to explain 11.3% on average. For instance, negative reviews in Yelp have 0.389 positive words per sentence, out of which 0.033 words follow a negation. This accounts for 8.7% of the usage of positive words. This suggests that negations before positive words only account for a small fraction of positive words, despite that they dominate other combinations of negations and sentiment lexicon. We hypothesize for positive words in negative reviews, they may be negated in ways beyond immediate preceding negations (e.g., "nor is the food great" and "fail to support").

Similarly, the number of negations followed by positive/negative words is a fraction of all the negations (14.2% in negative reviews and 9.7% in positive reviews). For example, "I will not return" counts as negation but there is no sentimental lexicon. We hypothesize that these negations also tend to express negative sentiments.

4.3 Sentence-level Sentiment Classification

To capture the sentiment of sentences with positive words or negations beyond negations immediately followed by positive words, we rely on sentiment classifiers. Specifically, we use sentence-level classification to quantify the extent of negative sentences in those contexts compared to the overall average in negative reviews.

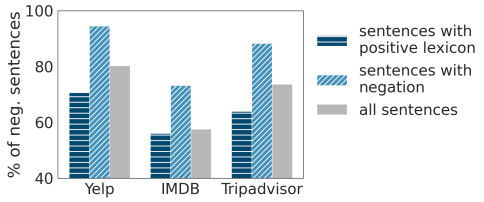
We fine-tune BERT (Devlin et al., 2019) to perform review-level classification for each dataset except SST and PeerRead. This is because all reviews in SST are very short and sentences in negative reviews are mostly negative whether negation occurs or not. In the case of PeerRead, the number of samples is too small to fine-tune the BERT model. For all other datasets except IMDB and Amazon Luxury Beauty, we randomly sample 25K positive reviews and 25K negative reviews as the training set, and 5K positive reviews and 5K negative reviews as the test set. For IMDB, we use 12.5K positive and 12.5K negative training samples provided for fine-tuning, and for Luxury Beauty, we use a balanced set of 2.3K positive and 2.3K negative samples for fine-tuning. We use 90% of the training samples to fine-tune the BERT model and 10% as the development set to select hyperparameters. We achieved accuracies varying from 94% to 98% for the test set reviews in all datasets. See the appendix for details of the data split and accuracies.

We use the BERT model fine-tuned on reviews to predict sentiment of sentences. Note that this prediction entails a distribution shift as sentences are shorter than full reviews used to fine-tune BERT models. However, this is a common strategy for evaluating rationales in the interpretable machine learning literature and there exists evidence that transformer-based models provide strong performance despite the distribution shift in the form of reduced inputs (DeYoung et al., 2020; Hsu et al., 2020; Carton et al., 2020).[5]

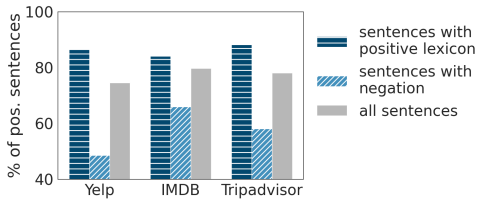
Figure 4a shows that sentences with positive words in negative reviews are more likely to be negative than to be positive (65.1% on average across all datasets; notably, IMDB is lower but still at 56.13%, above 50%).[6] It suggests that the majority of positive words are negated in some way. While the remaining minority of sentences with positive

[5] Bastan et al. (2020) investigates the reverse direction, i.e., from paragraph-level predictions to document-level predictions.

[6] Similar trends hold if we adjust the estimates using TPR, TNR, FPR, and FNR. See the appendix.



(a) Fractions of negative sentences in negative reviews.



(b) Fractions of positive sentences in positive reviews.

Figure 4: Sentence-level prediction results based on fine-tuned BERT classifiers. In negative reviews, sentences with positive words tend to be negative, and sentences with negations are overwhelmingly negative. In comparison, sentences with negations are more balanced (44.7% negative) in positive reviews.

words are indeed positive and align with the tendency to report positive views, our results highlight the important role of indirect expressions in explaining the positive words in negative reviews.

Furthermore, sentences with negation tend to be negative (88.6%) based on our classifiers, confirming our hypothesis that most negations are used to express negative sentiments in negative reviews. This is even higher than the average fraction of negative sentences (73.1%) among all sentences in negative reviews. In comparison, Figure 4b shows that positive words in positive reviews tend to reflect positive sentiments, indicating no common use of negation associated with positive words. Meanwhile, negations are not usually associated with negative sentiments in positive reviews (44.7%), substantially lower than negations associated with negative sentiments in negative reviews (88.6%).

5 Conclusion

In this paper, we investigate positivity bias in negative reviews and highlight the role of indirect expressions in understanding this phenomenon. We show that negations followed by positive words are more prevalent than any other combination in negative reviews. Given that these indirect wordings account for only 11.3% of the occurrences of positive words in negative reviews, we further show that such sentences with positive words tend to be negative, based on sentiment classifiers.

While our findings support the prevalence of indirect expressions, we do not take sentiment intensity into account. In practice, "not good" provides a different meaning from "not amazing". We believe exploring the relationship between negation and semantic intensity is a promising direction. Our lexical-driven approaches are limited by the lexicons included in the dictionaries, which are typically evaluated independent of the context, so their sentiment may be different in the specific context.[7] Similarly, our sentence-level prediction results are limited by the distribution shift when applying BERT trained on documents to sentences. It is reassuring that our high-level results hold across multiple datasets based on both lexical-driven approaches and sentence-level prediction.

As our study focuses on negative reviews in English, it is important to examine the generalizability of our results. For instance, it is important to understand to what extent the observed positivity bias in general expressions is driven by such indirect expressions. Another natural extension is to investigate other languages. Although our findings are limited to English reviews, we believe that they may be applicable to negative opinions in other languages, as Pollyanna hypothesis (Boucher and Osgood, 1969) has been validated across languages and cultures. Finally, our work has implications for sentiment-related applications in NLP. The prevalence of indirect expressions in negative reviews underscores the importance of modeling and understanding negation in sentiment analysis and sentiment transfer (Ettinger, 2020).

In general, we believe that online reviews not only provide valuable data for teaching machines to recognize sentiments but also allow us to understand how humans express sentiments. We hope that our work encourages future work to further investigate the framing choices when we express emotions and opinions, and their implications on NLP applications.

Acknowledgments

We thank anonymous reviewers and the members of the Chicago Human+AI lab for their helpful comments. This work was supported in part by an Amazon research award, a Salesforce research award, and NSF IIS-1941973.

[7] One reviewer pointed out an interesting hypothesis: judges assume the nicest interpretation of a word out of context in the annotation process, as a result, the Pollyanna hypothesis may be an instrumentation bias.

References

Adam A Augustine, Matthias R. Mehl, and Randy J. Larsen. 2011. A Positivity Bias in Written and Spoken English and Its Moderation by Personality and Gender. *Social Psychological and Personality Science*, 2(5):508–515. Publisher: SAGE Publications Inc.

Mohaddeseh Bastan, Mahnaz Koupaee, Youngseo Son, Richard Sicoli, and Niranjan Balasubramanian. 2020. Author's sentiment prediction. In *Proceedings of COLING*.

Jerry Boucher and Charles E Osgood. 1969. The pollyanna hypothesis. *Journal of verbal learning and verbal behavior*, 8(1):1–8.

Penelope Brown, Stephen C Levinson, and Stephen C Levinson. 1987. *Politeness: Some universals in language usage*, volume 4. Cambridge university press.

Alan Carr. 2011. *Positive psychology: The science of happiness and human strengths*. Routledge.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. In *Proceedings of EMNLP*.

Herbert L Colston. 1999. "not good" is "bad," but "not bad" is not "good": An analysis of three accounts of negation asymmetry. *Discourse Processes*, 28(3):237–256.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized NLP models. In *Proceedings of ACL*.

Peter Sheridan Dodds, Eric M Clark, Suma Desu, Morgan R Frank, Andrew J Reagan, Jake Ryland Williams, Lewis Mitchell, Kameron Decker Harris, Isabel M Kloumann, James P Bagrow, et al. 2015. Human language reveals a universal positivity bias. *Proceedings of the national academy of sciences*, 112(8):2389–2394.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

David Garcia, Antonios Garas, and Frank Schweitzer. 2012. Positive words carry less information than negative words. *EPJ Data Science*, 1(1):3.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Vera Hoorens. 2014. Positivity Bias. In Alex C. Michalos, editor, *Encyclopedia of Quality of Life and Well-Being Research*, pages 4938–4941. Springer Netherlands, Dordrecht.

Chao-Chun Hsu, Shantanu Karnwal, Sendhil Mullainathan, Ziad Obermeyer, and Chenhao Tan. 2020. Characterizing the value of information in medical notes. *Finding of EMNLP*.

Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. 2013. Unsupervised sentiment analysis with emotional signals. In *Proceedings of WWW*.

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of ICWSM*.

Naomi Kamoen, Maria BJ Mos, and Willem FS Dekker. 2015. A hotel that is not bad isn't good. the effects of valence framing and expectation in online reviews on text, reviewer and product appreciation. *Journal of Pragmatics*, 75:28–43.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Proceedings of NAACL*.

Isabel M Kloumann, Christopher M Danforth, Kameron Decker Harris, Catherine A Bliss, and Peter Sheridan Dodds. 2012. Positivity of the english language. *PloS one*, 7(1):e29484.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of ACL (system demonstrations)*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of EMNLP*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of ACL*.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

James W Pennebaker, Roger J Booth, and Martha E Francis. 2007. Linguistic inquiry and word count: Liwc [computer software]. *Austin, TX: liwc. net*, 135.

Christopher Potts. 2010. On the negativity of negation. In *Semantics and Linguistic Theory*, volume 20, pages 636–659.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of KDD*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of KDD*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of EMNLP*.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of ACL*.

Shusen Zhou, Qingcai Chen, and Xiaolong Wang. 2010. Active deep networks for semi-supervised sentiment classification. In *Proceedings of COLING*.

A Vader Lexicons

Table 3 shows the list of negation lexicons in Vader.

aint, arent, cannot, cant, couldnt, darent, didnt, doesnt, ain't, aren't, can't, couldn't, daren't, didn't, doesn't, dont, hadnt, hasnt, havent, isnt, mightnt, mustnt, neither, don't, hadn't, hasn't, haven't, isn't, mightn't, mustn't, neednt, needn't, never, none, nope, nor, not, nothing, nowhere, oughtnt, shant, shouldnt, uhuh, wasnt, werent, oughtn't, shan't, shouldn't, uh-uh, wasn't, weren't, without, wont, wouldnt, won't, wouldn't, rarely, seldom, despite

Table 3: Negation lexicons in Vader used for our negation analysis.

B Samples from PeerRead

Table 4 shows a list of 6 sentences with negation selected from random negative PeerRead reviews. Negations are mostly associated with positive words, both directly and indirectly.

Please do *not* make incredibly unscientific statements like this one :"
I'm *not* convinced about the value of having this artificial dataset.
For example, at the end of sec 4.4, " This result is *not* surprising, given that FOV-R contains additional information ....
It is *not* clear whether the improvements (if there is) of the ensemble disappear after data-augmentation.
Empirical analysis is *not* satisfactory.
But I'm *not* sure from reading the paper.

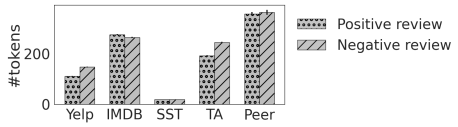
Table 4: Sentences with negation sampled from negative reviews of PeerRead. Positive words are in blue and negative words are in red. Negations are in italics.

C Additional Plots

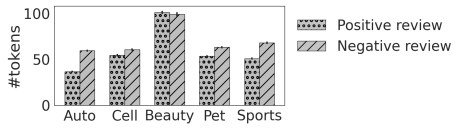
Length distribution. See Figure 5 for review-level length and Figure 6 for sentence-level length distribution for Amazon reviews.

Lexicon distribution. Figure 7 shows the sentiment lexicon distribution of all reviews using LIWC. Figure 8 shows the lexicon distribution of Amazon reviews using Vader.

Negation distribution. See Figure 9 and Figure 11 for the negation distribution of Amazon reviews using Vader and dependency parsing respectively. Figure 10 shows the negation distribution found using dependency parsing for non-Amazon reviews.



(a) SST, Yelp, IMDB, and Tripadvisor (non-Amazon datasets).



(b) Amazon datasets.

Figure 5: Review-level length distribution. This shows the length comparisons of positive and negative reviews of different datasets. The values represent the average number of tokens present in each review. Negative reviews are longer than positive reviews in all datasets except IMDB and Amazon Luxury Beauty.

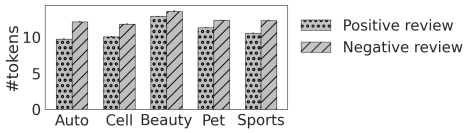
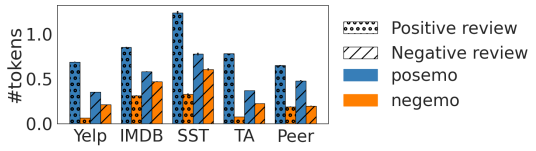
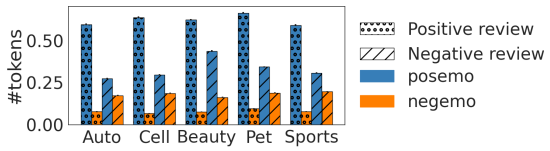


Figure 6: Sentence-level length distribution of Amazon datasets.



(a) Non-Amazon datasets.



(b) Amazon datasets.

Figure 7: Lexicon distributions based on LIWC. Figure 7a and Figure 7b shows the lexicon distribution of reviews using *posemo* and *negemo* LIWC categories. In all datasets, negative reviews have fewer positive emotions than positive reviews. They also have more positive words than negative words. This trend is similar to that obtained using Vader lexicons in case of non-Amazon reviews.

In case of negation distributions found using dependency parsing, we used Vader to identify positive and negative words.

Sentiment predictions. See Figure 4a and Figure 12 for the fractions of negative sentences in negative non-Amazon reviews measured by the BERT model. See Figure 13 for fractions of negative sentences in negative reviews of Amazon. Figure 14

Dataset	Training set	Validation set	Test set	Test accuracy (%)
Yelp	45000	5000	10000	97.51
IMDB	22500	2500	10000	94.38
Tripadvisor	45000	5000	10000	96.66
Automotive	45000	5000	10000	95.65
Cellphones and accessories	45000	5000	10000	95.39
Luxury beauty	4195	467	3040	96.10
Pet supplies	45000	5000	10000	95.60
Sports and outdoors	45000	5000	10000	95.12

Table 5: Dataset split and test accuracies of BERT fine-tuning. For all datasets except IMDB, Luxury Beauty, we use 45K samples as training set, 5K as validation set, and 10K as test set, randomly sampled from the entire dataset. In the case of IMDB, we use 22.5K samples for training and 2.5K samples for validation, randomly sampled from the provided training set of size 25K. We then use 10K samples randomly sampled from the provided test set of size 25K for testing purposes. In the case of Amazon Luxury Beauty, we use a balanced set of 4195 samples for training and 467 samples for validation. We then use 3K samples (imbalanced) randomly sampled from the dataset for testing. All these random samplings were done without replacement.

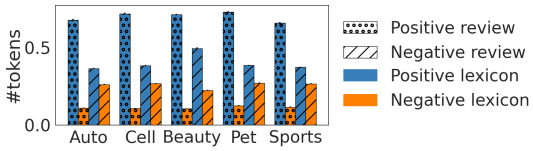
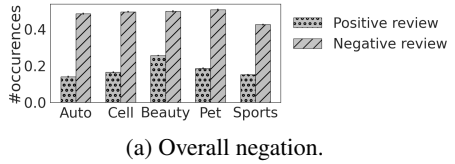
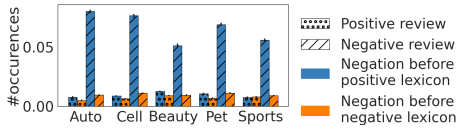


Figure 8: Lexicon distribution of Amazon datasets using Vader. Negative reviews have more positive words than negative words, similar to the trend in SST, Yelp, IMDB, Tripadvisor, and PeerRead.

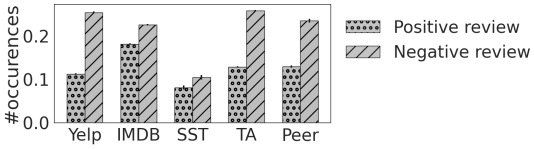


(a) Overall negation.

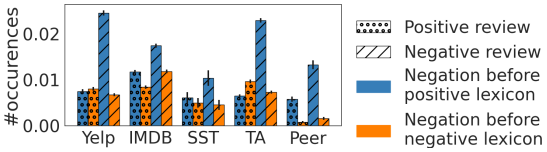


(b) Negation before positive and negative lexicons.

Figure 9: Negation distribution of Amazon datasets using Vader lexicons. Negative reviews use more negation words compared to positive reviews. Negative reviews have substantially more negation words associated with positive words than negative words.



(a) Overall negation.



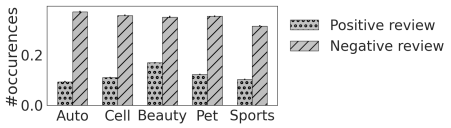
(b) Negation before positive and negative lexicons.

Figure 10: Negation distribution using dependency parsing - non-Amazon datasets. In all non-Amazon datasets, negative reviews use more negation words than positive reviews. This observation is inline with the negation results obtained using Vader lexicons. Dependency parsing is used to extract negations from reviews, and to identify words associated with a negation word.

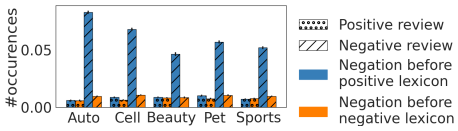
shows the fractions of positive sentences in positive reviews. Some of the fractions in our results are computed based on the TPR, TNR, FPR, and FNR of the BERT model. We used test set reviews of the datasets to compute these metrics as they give more accurate estimate of percentage of positive and negative sentences in reviews. All BERT classifiers that we used for predicting the sentiment of sentences are fine-tuned using the reviews of corresponding datasets. Table 5 shows the dataset split and test accuracies of BERT fine-tuning.

Hyperparameter tuning. We did hyperparameter tuning by varying number of epochs, batch size, and learning rate. We fine-tuned BERT for 4 epochs with batch sizes of 2, 4 and 8, with a learning rates of 1e-5 and 2e-5. Based on validation accuracies, the model trained for 2 epochs, with a batch size of 8 and learning rate of 2e-5 turned out to be the best performing model for most of the datasets.

Word-level results. Figure 15 shows the lexicon distribution using LIWC and Vader. See Figure 16 and Figure 17 for word-level results of negation distribution using Vader and dependency parsing respectively.



(a) Overall negation.



(b) Negation before positive and negative lexicons.

Figure 11: Negation distribution using dependency parsing - Amazon datasets. Figure 11a shows that negative reviews have substantially more negation words than positive words. Figure 11b shows the negation distribution associated with positive and negative words. This corresponds to about 16.68% of all negation words used in the positive and negative reviews based on our dictionary. Negative reviews also have substantially more negations before positive words, compared to other combinations.

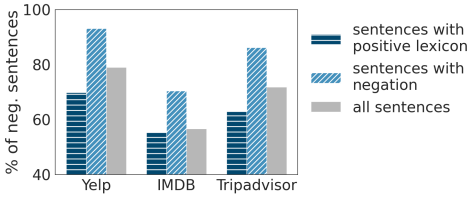
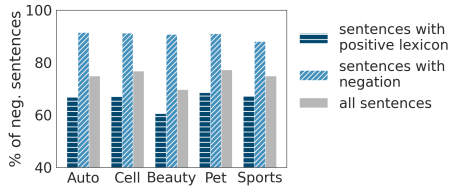
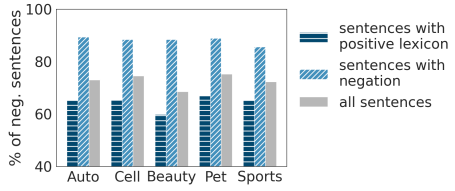


Figure 12: Fractions of negative sentences in negative reviews of Yelp, IMDB, and Tripadvisor. These fractions are corrected using TPR, TNR, FPR, and FNR. It can be seen that higher proportion of negative reviews with negation are classified as negative by our BERT model. This shows that negations in negative reviews are mostly used to express negative opinions. This observation holds for other datasets also.

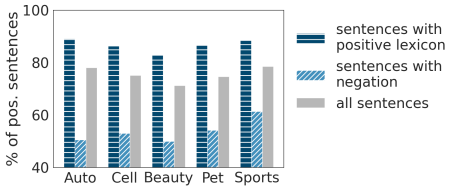


(a) Fractions based on accuracy.

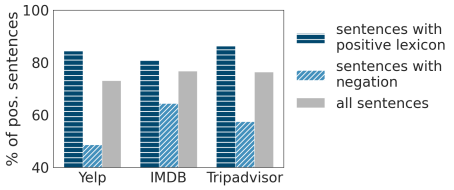


(b) Fractions based on TPR, TNR, FPR, and FNR.

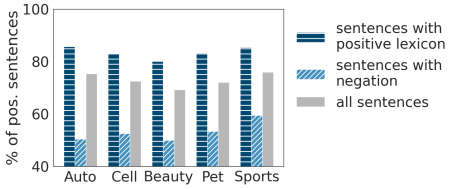
Figure 13: Fractions of negative sentences in negative Amazon reviews based on fine-tuned BERT classifiers. The distribution confirms our hypothesis that most negations are used to express negative sentiments.



(a) Fractions based on accuracy.

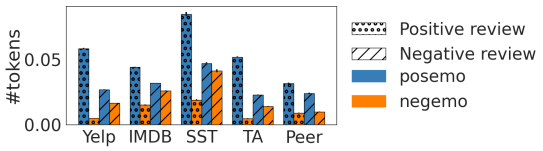


(b) Fractions based on TPR, TNR, FPR, and FNR.

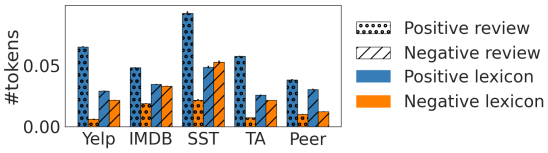


(c) Fractions based on TPR, TNR, FPR, and FNR.

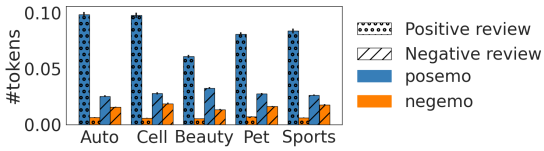
Figure 14: Fractions of positive sentences in positive reviews. We can see that negations in positive reviews are more balanced with positive and negative sentences when compared to negative reviews. Also, sentences with positive lexicons are mostly positive (86.5%). There are very few negative sentences with positive lexicons. This holds for all datasets.



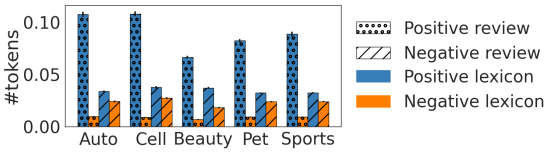
(a) LIWC distribution of non-Amazon datasets.



(b) Vader distribution of non-Amazon datasets.

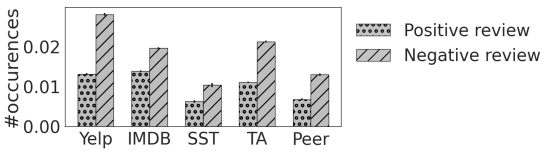


(c) LIWC distribution of Amazon datasets.

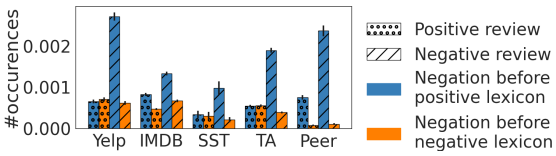


(d) Vader distribution of Amazon datasets.

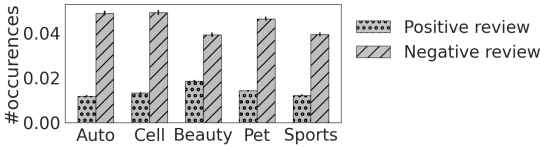
Figure 15: Word-level lexicon distribution. At the word-level, positive reviews have more positive words than negative reviews. However, negative reviews contain more positive words than negative words (except SST with Vader). The trend that we observe in the sentence-level results can be seen here as well.



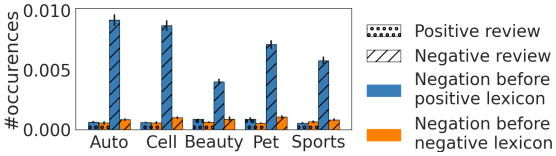
(a) Non-Amazon datasets.



(b) Non-Amazon datasets.

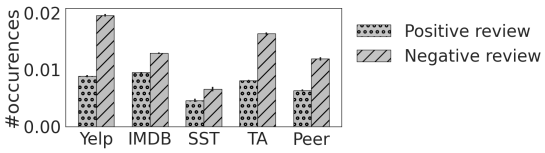


(c) Amazon datasets.

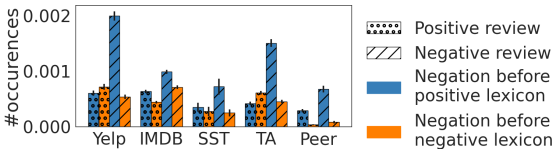


(d) Amazon datasets.

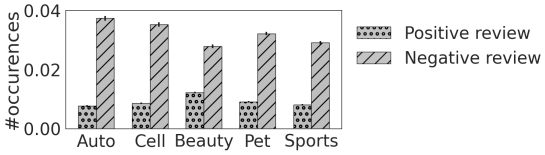
Figure 16: Word-level negation distribution using Vader. Figure 16a and Figure 16c indicate the more frequent use of negation in negative reviews than in positive reviews at the word-level. Negative reviews have more negations before positive words in all datasets. This difference is substantially large in case of Yelp, Tripadvisor, PeerRead and Amazon reviews. This shows that although negative reviews have more positive words than negative words, these positive words are associated with negations.



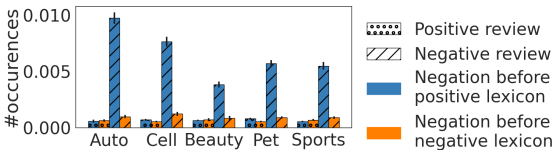
(a) Overall negation in non-Amazon datasets.



(b) Negation before positive and negative lexicons in non-Amazon datasets.



(c) Overall negation in Amazon datasets.



(d) Negation before positive and negative lexicons in Amazon datasets.

Figure 17: Word-level negation distribution of all reviews using dependency parsing. With dependency parsing, we observe the same pattern as in Figure 16. Negative reviews in Yelp, Tripadvisor, PeerRead and Amazon datasets have substantially more negations in general and also before positive words. This high number of negation associated with positive words partially explains the higher proportion of positive words in negative reviews.

PRAL: A Tailored Pre-Training Model for Task-Oriented Dialog Generation

Jing Gu* UC Davis jkgu@ucdavis.edu
Qingyang Wu* Columbia University qingyang.wu@columbia.edu
Chongruo Wu UC Davis crwu@ucdavis.edu
Weiyan Shi Columbia University shi.weiyan@columbia.edu
Zhou Yu Columbia University zy2461@columbia.edu

Abstract

Large pre-trained language generation models such as GPT-2 have demonstrated their effectiveness as language priors by reaching state-of-the-art results in various language generation tasks. However, the performance of pre-trained models on task-oriented dialog tasks is still under-explored. We propose a Pre-trained Role Alternating Language model (PRAL), explicitly designed for task-oriented conversational systems. We design several techniques: start position randomization, knowledge distillation, and history discount to improve pre-training performance. In addition, we introduce a high quality large-scale task-oriented dialog pre-training dataset. We effectively adapt PRAL on three downstream tasks. With much less training data, PRAL outperforms or is on par with state-of-the-art models.

1 Introduction and Related Work

Current approaches to building task-oriented dialog systems still require a substantial amount of annotations and therefore are labor-intensive. On the other hand, large-scale pre-trained language models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2019) have achieved great success on various NLP tasks. There have been several attempts to apply these language models to dialog systems directly. For example, Transfer-Transfo (Wolf et al., 2019) fine-tuned GPT on the Persona-Chat dataset (Zhang et al., 2018b) and achieved the state-of-the-art performance on chitchat dialog generation. DialoGPT (Zhang et al., 2020) utilizes a large Reddit corpus to further pre-train GPT-2 (Zhang et al., 2020). All of these studies pointed to a promising direction towards building dialog systems with large-scale language models and less annotation.

However, these language models applied to dialog systems still have some limitations. First, further pre-training language models for dialog systems requires a considerable amount of training data. Small pre-training dialog datasets would not provide a large amount of commonsense knowledge needed for dialog generation. However, a diverse collection of high-quality dialog datasets is difficult to obtain. Besides, these language models usually do not consider dialog feature in their structures.

To tackle these issues, we propose Pre-trained Role Alternating Language model (PRAL), a language model designed explicitly for dialog generation. To begin with, we collect and process 13 dialog datasets, ranging from TV transcripts to pizza ordering dialogs, to enrich the pre-training data with high-quality dialog corpora. Second, we adopt ARDM proposed in Wu et al. (2019) and use two separate GPT-2 to model the two speakers in the dialog. Next, we apply Start Position Randomization (SPR) to cope with the variable lengths in dialogs, which prevents the language model from binding the position index with the text information. Additionally, we utilize a Teacher model to perform knowledge distillation and incorporate common sense knowledge into the dialog generation. Finally, we re-weigh each utterance with discount factors and emphasize on the later part in a dialog to better incorporate contextual information.

In summary, we propose PRAL and design several effective techniques to improve the dialog model pre-training. Our pre-trained model improves the success rate on CamRest676 and MultiWOZ dataset, and the coherence and diversity scores on PersuasionForGood. Our model is data-efficient and use 10x less than SOLOIST and 1000x less than DialoGPT in terms of training data size. We also process and present a collection of high-

* Equal contribution

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 305–313
August 1–6, 2021. ©2021 Association for Computational Linguistics

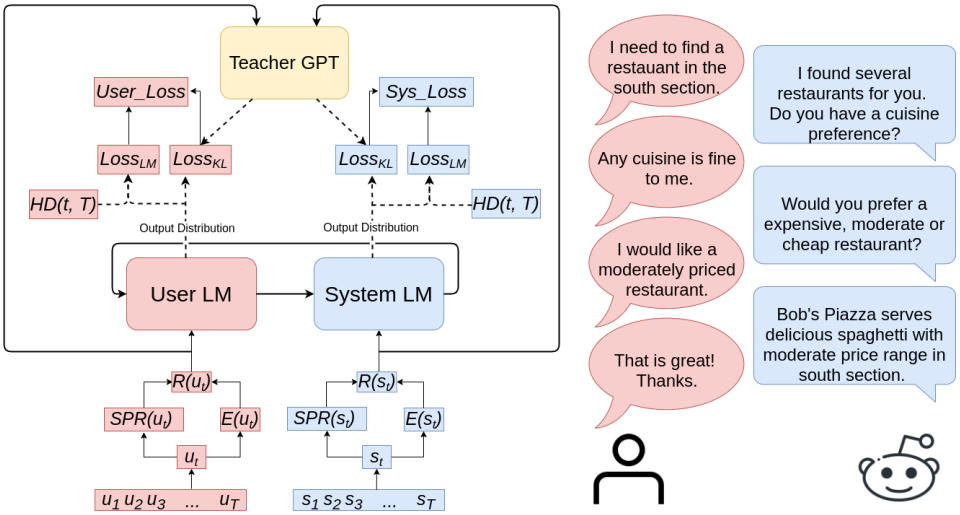


Figure 1: An overview of PRAL's architecture. PRAL has separate language models for each speaker. The representation of user utterance u_t or system u_s is from word embedding E and the randomized position embedding SPR. $HD(t, T)$ is the history discount weight of each utterance. Teacher GPT provides supervision for the two language models. $Loss_{LM}$ and $Loss_{KL}$ denote the losses of the language models and the KL divergence.

Dataset Statistics	
# Datasets	13
# Dialogues	142,298
Avg. turns per dialogue	12.66
Avg. tokens per turn	11.78
Avg. tokens per dialogue	149.25
Total unique tokens	108,106

Table 1: Statistics of our dataset

quality dialog datasets suitable for pre-training large-scale language models for dialog systems.

2 *PretrainDial* Dataset for Pre-training

Large clean dialog datasets are difficult to find. Therefore, we constructed *PretrainDial*, a large-scale multi-domain dialog corpus for dialog pre-training. *PretrainDial* is a large-scale pre-training dataset and can only be collected from existing dialogs. We carefully selected 13 existing dialog corpora, ranging from chitchat such TV transcripts to task-oriented dialogs, and design a sophisticated text processing pipeline. Table. 1 shows the statistics of the *PretrainDial* dataset. Please check appendix A for more details about the dataset statistic and the text processing pipeline.

3 Methods

In this section, we will first briefly introduce ARDM, our base dialog model, and then describe a set of techniques proposed in PRAL. Figure 1 shows the main structure of PRAL.

3.1 Alternating Roles Dialog Model

The basic idea behind ARDM (Wu et al., 2019) is to simultaneously model the user and system with two separate GPT-2 to capture the different language styles among different speakers. A dialog can be considered as a sequence of utterances $d = \{u_1, s_1, u_2, s_2, \ldots, u_T, s_T\}$, where T is the total number of turns. We use p_u and p_s to represent the probability of the user utterance and system utterance. The dialog distribution is defined as:

$$p(d) = \prod_{t=1}^{T} p_u(u_t|u_{<t}, s_{<t})\, p_s(s_t|u_{\leq t}, s_{<t}) \quad (1)$$

However, ARDM does not contain prior knowledge about dialog. In contrast, PRAL is designed for dialog system and absorbs abundant dialog knowledge during the pre-training process. To further improve ARDM or other dialog generation models, we propose three effective techniques to improve pre-training efficiency.

3.2 Start Position Randomization

We propose to randomize the start position to improve pre-training model's quality. Transformer-based language models use position embedding to encode the location information for each token. It supports a fixed maximum position, and the position index always starts from 0. However, since most dialogs contain less than 1024 tokens, most vectors in the positional embedding would remain

zero and not update during pre-training. Besides, position embedding should only provide location information. However, the fixed start position will bond specific text with a particular position index. For example, "hi" is bonded with index 0 as "hi" usually appears at the beginning of the dialog. Therefore, the model is likely to overfit on the first several positional embeddings.

To address this issue, we propose to perform Start Position Randomization (SPR). L stands for the total number of tokens in a dialog, and the maximum start position index is $1024 - L$. We randomize the start position to be any number between 0 to $1024 - L$. It would disentangle the positional information from the textual meaning and force the model to update all the positional embeddings.

3.3 Teacher GPT

Neural networks models suffer from catastrophic forgetting (Kirkpatrick et al., 2016). Since we have finetuned GPT-2 with a new dialog corpus, the updated model is at risk in forgetting the prior knowledge from the original GPT-2. Teacher Model is used to calculate the distillation loss (Hinton et al., 2015) between the fixed GPT and our two language models. It constrains the language model from generating a token distribution that is too different from the original token distribution. The Teacher Model has two functions. First it avoids language model from catastrophic forgetting the knowledge in the original GPT-2 weights (Kirkpatrick et al., 2016). Secondly, when GPT-2 Large is used as Teacher Model, it imparts knowledge into our language models. The ablation study in table 2a validates the the functions.

3.4 History Discount

In dialog generation, historical utterances closer to current utterance should have a more significant impact on the generation than the ones that are further. Because in human conversations, we tend to prioritize local coherence over distant history coherence as well .Therefore, we introduce discount factor γ to re-weigh the importance of each utterance based on the turn number. For a dialog with a total number of T utterances and its current utterance index to be t, we weigh the language model loss with γ^{T-t}. By incorporating the discount factor γ, the model focus more on recent history in the generation process.

3.5 Optimization

We use a language modeling loss to optimize our model, shown in Equation 2.

$$Loss_{LM} = \sum_{t=1}^{T} \gamma^{T-t} \sum_{l=1}^{L_t - 1} CE(P_{tl},\, G_{t(l+1)}) \tag{2}$$

CE denotes the cross-entropy loss. T is the total number of utterances in a dialog, and L_t is the total number of tokens in the t^{th} utterance. For the loss of each utterance t in the dialog, it is weighed by the discount factor described in section 3.4. We combine loss from all words as the cross-entropy between the output probability distribution $P_{t(l+1)}$ and the ground truth $G_{t(l+1)}$.

The final loss combines the language model loss and KL divergence loss:

$$Loss = Loss_{LM} + \alpha\, \mathrm{KL}(p,\, p^{constriant}) \tag{3}$$

The factor α is used to expedite model convergence and it decreases exponentially as the number of iterations increases, i.e. $\alpha = \alpha_0\, \lambda^{iter}$.

4 Experiments

We pre-train PRAL on *PretrainDial*. We use GPT-2 large as the Teacher model. We use AdamW optimizer with warm-up steps as 10 percent of the training step. The learning rate is set to be 1×10^{-4}. For the calculation of loss, we set α_0 to be 0.1 and set λ to be 0.9999. The discount factor γ is set to be 0.95. To show the generalizability, we finetune PRAL on three downstream dialog generation tasks, CamRest676, MultiWOZ and PersuasionForGood, as is shown in Table. 2. Refer to Appendix B for more experiment setting.

CamRest676 (Rojas-Barahona et al., 2016) is a dialog dataset for restaurant recommendation containing 680 dialogs. We use BLEU-4 metrics to measure the quality of generated sentences, and Success F1 to evaluate the responses on specific slots, such as address, phone, postcode. Sequicity is the state-of-the-art method in task-oriented dialog tasks that requires annotations. PRAL beat all other models, including a concurrent work SOLOIST (Peng et al., 2020) on both BLEU-4 and Success F1. It is worth noting that PRAL does not need any annotation. SOLOIST and DialoGPT have a close performance with our model. However, SOLOIST uses around 1 Million dialogues, DialoGPT uses around 147 million dialogues, meanwhile we only use around 142K dialogues, which

Model	BLEU-4	Success F1
Sequicity	21.4	0.852
Sequicity (w/o RL)	22.9	0.821
GPT-2-finetune	21.8	0.851
DialoGPT	25.2	0.861
SOLOIST	25.5	0.871
ARDM	26.2	0.864
PRAL	**27.2**	**0.874**
- w/ Teacher GPT(small)	26.9	0.869
- w/o Teacher GPT	25.0	0.865
- w/o loss discount	27.0	0.867
- w/o SPR	26.6	0.869

(a) Results on CamRest676 dataset.

Model	Supervision		BLEU-4	Inform	Success
	Dialog State	Dialog Act			
Human	-	-	-	0.989	0.965
Baseline	✓	×	18.9	0.825	0.729
HDSA	✓	✓	**23.6**	0.877	0.734
LaRL	✓	×	12.8	0.828	0.792
SOLOIST	✓	×	18.0	0.896	0.793
ARDM	×	×	20.6	0.874	0.728
PRAL	×	×	21.2	**0.899**	**0.798**

(b) Results on MultiWOZ dataset

	Perplexity ↓	BLEU-1 ↑	BLEU-2 ↑	Fluency ↑	Logic ↑	Coherence ↑	Diversity ↑	Overall ↑	Avg.Donation ↑
ARDM	**10.1**	16.5	6.44	0.39	0.41	0.37	0.27	0.18	0.62
PRAL	10.3	**17.3**	**10.9**	**0.61**	**0.59**	**0.63**	**0.73**	**0.82**	**0.99**

(c) PersuasionForGood. Automatic Evaluation and Human Evaluation Results

Table 2: Evaluation on three datasets

is a thousand times less. This further shows PRAL is data-efficient.

Ablation studies on CamRest676 shows that the Teacher GPT plays the most important role. The fact that PRAL with Teacher GPT (Small) in table 2a outperforms PRAL without Teacher GPT (Small) shows the importance of the knowledge in the original model weights. When using GPT-2 Large as Teacher Model, the performance is better than that of using GPT-2 small, which validates the effect of knowledge distillation.

MultiWOZ (Budzianowski et al., 2018) contains around 10k dialogues covering various domains. We evaluate the models with on BLEU-4, Inform Rate, and Success Rate which measures if the system provides the requested information. PRAL outperforms the attention seq2seq model which is used as the baseline in Multiwoz (Budzianowski et al., 2018) in all metrics. Without using any annotation, PRAL also outperforms or achieve comparable results with HDSA (Budzianowski et al., 2018), LaRL (Zhao and Kawahara, 2019) and SOLOIST. Except for HDSA which requires both dialog state and dialog act, PRAL achieves a better BLEU score than all other models. PRAL outperforms ARDM in all metrics, which further validates the effectiveness of the pre-training process.

PersuasionForGood We also evaluate our method on PersuasionForood (Wang et al., 2019), where a persuader tries to persuade users to donate money to children. There are a total of 1,017 dialogues. Although not a traditional task-oriented dialog benchmark, it is a good benchmark for human evaluation. Automatic metrics evaluation is efficient but could fail to capture the text quality on a deeper and complicated level. We choose this task also because it benefits children. Unlike CamRest676 and Multiwoz, the language in PersuasionForGood dataset is so diverse that BLEU-4 scores of all of the models are too low to be scientific metrics. Therefore, we use BLEU-1 and BLEU-2 instead. Our model achieves a significantly higher score on BLUE metrics, especially on BLEU-2 (63% up). In human evaluation, we ask evaluators that how much they are willing to donate after the conversation and acquire their ratings in terms of fluency, logic, coherence, and diversity. The result suggests that PRAL outperforms ARDM on all the metrics. For human evaluation details, please refer to Appendices C.

Case studies show some linguistic problems in ARDM, such as repetition and unnaturalness. Meanwhile, with pre-training, PRAL is more natural and persuasive. Please refer to Appendices D for an example of PRAL and ARDM.

5 Conclusion

We propose PRAL, a large pre-trained dialog system for task-oriented generation. We incorporated methods that are designed for large dialog system into PRAL with good performances on three downstream tasks. The model generates more fluent, coherent, diverse, and logical dialogs according to human evaluation results. We also release a high-quality dialog dataset for the pre-training process.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525, Hong Kong, China. Association for Computational Linguistics.

Chit-Chat Challenge. a. Chitchat dataset. https://github.com/BYU-PCCL/chitchat-dataset.

Kaggle Challenge. b. Friends corpus. https://www.kaggle.com/vinayvk/friends-series-data-set.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *CMCL@ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.

Joachim Fainberg, Ben Krause, Mihai Dobre, Marco Damonte, Emmanuel Kahembwe, Daniel Duma, Bonnie L. Webber, and Federico Fancellu. 2018. Talking to myself: self-dialogues as data for conversational agents. *ArXiv*, abs/1809.06641.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *IJCNLP*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached conversational preference elicitation: A case study in understanding movie preferences. In *SIGDIAL 2019*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *ArXiv*, abs/1909.05855.

Reddit. Reddit corpus. https://zissou.infosci.cornell.edu/convokit/documentation/subreddit.html.

Lina Maria Rojas-Barahona, Milica Gaić, Nikola Mrksic, Pei hao Su, Stefan Ultes, Tsung-Hsien Wen, Steve J. Young, and David Vandyke. 2016. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *ACL*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019. Alternating roles dialog model with large-scale pre-trained language models.

Justine Zhang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. 2018a. Conversations gone awry: Detecting early signs of conversational failure. In *ACL*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*.

Tianyu Zhao and Tatsuya Kawahara. 2019. Effective incorporation of speaker information in utterance encoding in dialog.

A Dataset sources

Our dataset contains high-quality dialogues which are selected from other 13 datasets listed in Table 3. PretrainDial is a large-scale pre-training dataset and can only be collected from existing dialogs. Due to the page limit as a short paper, we didn't elaborate on the process in the paper. First, we collected dialog datasets that are commonly used in recent years. Then we filtered out the datasets with various standards such as content appropriateness. For example, we filtered "Conversations Gone Awry" Dataset because the conversation involves necessary background knowledge. Then, we process the text in the selected datasets. This step is essential since these datasets contain unnecessary noise, especially for datasets that contain raw text such as Friends dataset. The processing includes: (1) We replaced less informative appeared entity. For example, replace a long URL link with the word "URL". (2) Delete meaningless repetition. (3) Delete responses that are not written in English. (4) Delete offensive language. (5).In some datasets such as Reddit, the conversation involves more than two people, so we extract a complete conversation flow involving only two people. Note there are more detailed process steps. We cannot describe all of them. We will release the text processing script, which we believe is helpful for the community when collecting dialog datasets.

B Experiment Setting Detail

B.1 Training Details

We initialize PRAL with a large pre-trained language model GPT-2 small with 117M parameters (Radford et al., 2019). We follow a special format in GPT-2 as the "trigger" so that the model can zero-shot dialog response generation, by prefixing the user role token "A:" or "B:", and suffixing the end of utterance token "\n\n\n". We first pre-train PRAL on *PretrainDial* and then further fine-tune PRAL on the specific task dataset. We apply AdamW optimizer (Loshchilov and Hutter, 2019), and the number of warm-up ratio is set to 0.1. Learning rate is 1×10^{-4} in the pre-training process and 3×10^{-5} in fine-tune process.The dropout rate is set to 0.1 for all tasks. For the calculation of loss in the pre-training process, we set α_0 to be 0.1 and set λ to be 0.9999. The discount factor γ is set to be 0.95.

B.2 Decoding Details

In the downstream task, we decode utterances by nucleus sampling (Holtzman et al., 2020) with different hyper-parameters (top-p, top-k). We also vary the temperature of $T < 1$ to find the best setting for the specific down-stream dialog task. We use nucleus sampling for all methods. In CamRest676 task, we set top-p 0.2 and temperature 0.7 for our model. For MultiWOZ task, we set the top-p to 0.2 and the temperature to 0.7. In PersuasionForGood task, to generate diverse responses, we use a top-p of 0.9 and a temperature of 0.7.

C Human Evaluation Detail

Twenty people participated in the human evaluation. ARDM is the state-of-the-art model for PersuasionForGood task. Each person will have ten conversations with PRAL and ARDM in random orders, five conversations for each model. 1) For the donation task, the participants will be asked, "How much will you donate after talking to the bot? Please choose from 0-2$". 2) For fluency, logic, and coherence, the participants will be asked, "Which one do you think is more fluent/logic/coherent?" and choose the model with better performance on the corresponding metric. 3) For diversity, participants compare performances across dialogs, so they will be asked, "Which bot do you think generates more diverse responses?" after talking with each model for five times.

D Persuasion Example

Table 4 and Table 5 show an PRALexample and an ARDM example.

E Ethical Statement

As knowledge in language model comes from training corpus, we used a strict pipeline to avoid offensive and other inappropriate text. We believe such a selective training source could contribute a more unbiased dialog system. However, the Teacher Model could be a potential source of biased knowledge, although we notice that is not the case in our experiments. In the future, we will also explore how to select the knowledge to import from Teacher Model.

1	CCPE-M dataset (Radlinski et al., 2019)
2	ChitChat-Dataset (Challenge, a)
3	Conversations Gone Awry Dataset(Wiki) (Zhang et al., 2018a)
4	Conversations Gone Awry Dataset(CMV) (Zhang et al., 2018a)
5	Cornell Movie-Dialogs Corpus (Danescu-Niculescu-Mizil and Lee, 2011)
6	DailyDialog (Li et al., 2017)
7	Frames Dataset (El Asri et al., 2017)
8	PersonaChat ConvAI2 Dataset (**?**)
9	The Schema-Guided Dialogue Dataset (Rastogi et al., 2019)
10	Self-dialogue Corpus (Fainberg et al., 2018)
11	Friends Series Dataset (Challenge, b)
12	Taskmaster-1 (Byrne et al., 2019)
13	Reddit Corpus (Reddit)

Table 3: 13 Dataset Sources

Role	**Utterance**
SYS	Good morning, how are you doing today?
USR	Good!
SYS	Have you ever heard of Save the Children?
USR	I have not heard of it. Would you like to give an introduction?
SYS	It is a charity that helps children in need around the world.
USR	Cool
SYS	I would like to give a little bit more information.
USR	Sure. Go ahead.
SYS	Save the Children is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries.
USR	It is a good organization. I am glad to know it.
SYS	Yes, they are a very professional organization. Would you be interested in donating some of your task money to this charity?
USR	Sure. I would like to donate some money to it.
SYS	Thank you so much!
USR	no problem.
SYS	How much do you like to donate to the charity now? Your donation will be directly deducted from your task payment.
USR	I would like to give $2 to them.
SYS	Thank you so much!
USR	You are welcome
SYS	I hope you have a great day!
USR	You too.

Table 4: An example conversation between a machine persuader trained by PRAL and a human persuadee.

Role	Utterance
SYS	Good morning.
USR	Good!
SYS	Have you ever heard of Save the Children?
USR	I have not heard of it. Would you like to give an introduction?
SYS	It is a charity that helps children.
USR	Cool
SYS	Do you wanna know more about it?.
USR	Sure. Go ahead.
SYS	It is a charity that helps children.
USR	I know. But what is this organization really about?
SYS	Save the Children is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries.
USR	It is a good organization. I would like to make some contributions.
SYS	Would you be interested in donating some money to this charity?
USR	Sure. I think I can do that.
SYS	How much do you like to donate to the charity now? Your donation will be directly deducted from your task payment.
USR	I would like to give $1 to them.
SYS	Thank you so much! Bye!
USR	Bye!

Table 5: An example conversation between a machine persuader trained by ARDM and a human persuadee.

ROPE: Reading Order Equivariant Positional Encoding for Graph-based Document Information Extraction

Chen-Yu Lee[†], **Chun-Liang Li**[†], **Chu Wang**[§,*], **Renshen Wang**[‡], **Yasuhisa Fujii**[‡], **Siyang Qin**[‡], **Ashok Popat**[‡], **Tomas Pfister**[†]
[†]Google Cloud AI, [§]McGill University, [‡] Google Research
[†,‡]{chenyulee, chunliang, rewang, yasuhisaf, qinb, popat, tpfister}@google.com
[§]chu.wang@mail.mcgill.ca

Abstract

Natural reading orders of words are crucial for information extraction from form-like documents. Despite recent advances in Graph Convolutional Networks (GCNs) on modeling spatial layout patterns of documents, they have limited ability to capture reading orders of given word-level node representations in a graph. We propose Reading Order Equivariant Positional Encoding (ROPE), a new positional encoding technique designed to apprehend the sequential presentation of words in documents. ROPE generates unique reading order codes for neighboring words relative to the target word given a word-level graph connectivity. We study two fundamental document entity extraction tasks including word labeling and word grouping on the public FUNSD dataset and a large-scale payment dataset. We show that ROPE consistently improves existing GCNs with a margin up to 8.4% F1-score.

1 Introduction

Key information extraction from form-like documents is one of the fundamental tasks of document understanding that has many real-world applications. However, the major challenge of solving the task lies in modeling various template layouts and formats of documents. For example, a single document may contain multiple columns, tables, and non-aligned blocks of texts (e.g. Figure 1).

The task has been studied from rule-based models (Lebourgeois et al., 1992) to learning-based approaches (Palm et al., 2017; Tata et al., 2021). Inspired by the success of sequence tagging in NLP (Sutskever et al., 2014; Vaswani et al., 2017; Devlin et al., 2019), a natural extension is applying these methods on linearly serialized 2D documents (Palm et al., 2017; Aggarwal et al., 2020).

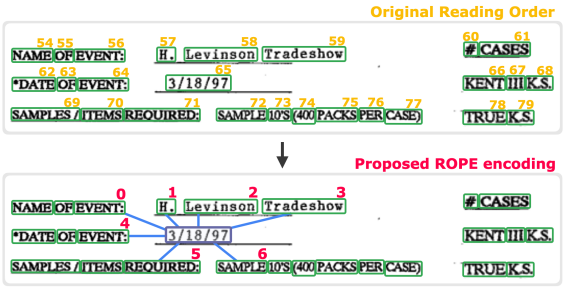


Figure 1: Illustration of the proposed Reading Order Equivariant Positional Encoding (ROPE). **Top:** a portion of a form document with the original word reading order. **Bottom:** given a graph connectivity, ROPE generates equivariant reading order codes with respect to the target word (in this case the date "3/18/97").

Nevertheless, scattered columns, tables, and text blocks in documents make the serialization extremely difficult, largely limiting the performance of sequence models. Katti et al. (2018); Zhao et al. (2019) explore to directly work on 2D document space using grid-like convolutional models to better preserve spatial context during learning, but the performance is restrictive to the resolution of the grids. Recently, Qian et al. (2019); Davis et al. (2019); Liu et al. (2019) propose to represent documents using graphs, where nodes define word tokens and edges describe the spatial patterns of words. Yu et al. (2020) show state-of-the-art performance of Graph Convolutional Networks (GCNs) (Duvenaud et al., 2015) on document understanding.

Although GCNs capture the relative spatial relationships between words through edges, the specific word ordering information is lost during the graph aggregation operation, in the similar way to the average pooling in Convolutional Neural Networks (CNNs). However, we believe reading orders are strong prior to comprehending languages. In this work, we propose a simple yet effective Reading Order Equivariant Positional Encoding (ROPE) that embeds the relative reading order context into

[*] Work done while an intern at Google Research.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 314–321
August 1–6, 2021. ©2021 Association for Computational Linguistics

graphs, bridging the gap between sequence and graph models for robust document understanding. Specifically, for every word in a constructed graph, ROPE generates unique reading order codes for its neighboring words based on the graph connectivity. The codes are then fed into GCNs with self-attention aggregation functions for effective relative reading order encoding. We study two fundamental entity extraction tasks including word labeling and word grouping on the public FUNSD dataset and a large-scale payment dataset. We observe that by explicitly encoding relative reading orders, ROPE brings the same or higher performance improvement compared to spatial relationship features in existing GCNs in parallel.

2 Other Related Work

Attention models show state-of-the-art results in graph learning (Veličković et al., 2018) and NLP benchmarks (Vaswani et al., 2017). As attention models with positional encodings are proven to be universal approximators of sequence-to-sequence functions (Yun et al., 2020), encoding positions or ordering is an important research topic. For sequence, learned positional embeddings (Gehring et al., 2017; Devlin et al., 2019; Shaw et al., 2018), sinusoidal functions and its extensions (Liu et al., 2020) have been studied. Beyond that, positional encodings are explored in graphs (You et al., 2019), 2D images (Parmar et al., 2018) and 3D structures (Fuchs et al., 2020). Lastly, graph modeling is also applied to other document understanding tasks, including document classification (Yao et al., 2019) and summerization (Yasunaga et al., 2017).

3 Method

We follow recent advances in using GCNs for document information extraction that relax any serialization assumptions by sequence modeling. GCNs take inputs (word tokens in this case) of arbitrary numbers, sizes, shapes and locations, and encode the underlying spatial layout patterns of documents through direct message passing and gradient updates between input embedding in the 2D space.

Node definition. Given a document D with N tokens denoted by $T = \{t_1, t_2, ..t_N\}$, we refer t_i to the i-th token in a linearly serialized text sequence returned by the Optical Character Recognition (OCR) engine. The OCR engine generates the bounding box sizes and locations for all tokens, as

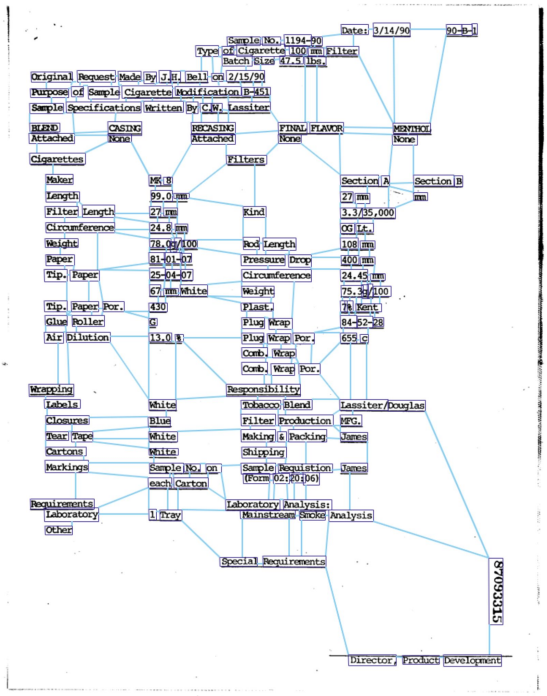


Figure 2: Sample of a β-skeleton graph of a document of FUNSD.

well as the text within each box. We define node input representation for all tokens T as vertices $V = \{v_1, v_2, ..v_N\}$, where v_i concatenates quantifiable attributes available for t_i. In our design, we use two common input modalities: (a) word embeddings from an off-the-shelf pre-trained BERT model (Devlin et al., 2019), and (b) spatial embeddings from normalized bounding box heights, widths, and Cartesian coordinate values of four corners.

Edge definition. While the vertices V represent tokens in a document, the edges characterize the relationship between the vertices. Precisely, we define directional edges for a set of edges E, where each edge e_{ij} connects two vertices v_i and v_j, concatenating quantifiable edge attributes. In our design, we use two input modalities given an edge e_{ij} connecting two vertices: (a) spatial embeddings from horizontal and vertical normalized relative distances between centers, top left corners and bottom right corners of the bounding boxes. It also contains height and width aspect ratios of v_i, v_j, and relative height and width aspect ratios between v_i and v_j. (b) Visual embeddings that utilizes ImageNet pre-trained MobileNetV3 (Howard et al., 2019) to extract visual representations of union bounding boxes containing v_i and v_j. The visual embedding in edge formation picks up visual cues

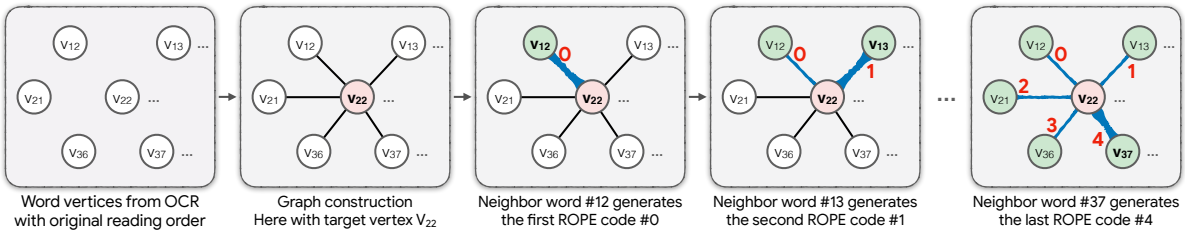


Figure 3: Implementation of the proposed Reading Order Equivariant Positional Encoding (ROPE). Given a graph connectivity, ROPE iterates through the neighboring word vertices in the original reading order and assigns new ROPE codes (red numbers) to the neighbors, starting from zero. Note that the proposed ROPE codes remain unchanged if the neighbors and the target shift equally in the document with the same relative reading order, therefore being equivariant.

such as colors, fonts, separating symbols or lines between two token bounding boxes (through their union bounding box). We refer to the spatial embedding in (a) as the edge geometric (EdgeGeo) feature used in the experimental section.

Graph construction. Our implementation is based on the β-skeleton graph (Kirkpatrick and Radke, 1985) with $\beta = 1$ for graph construction. By using the "ball-of-sight" strategy, β-skeleton graph offers high connectivity between word vertices for necessary message passing while being much sparser than fully-connected graphs for efficient forward and backward computations (Wang et al., 2021). A β-skeleton graph example can be found in Figure 2, and more can be found in Figure 5 in the Appendix.

Aggregation function. Inspired by the Graph Attention Networks (Veličković et al., 2018) and the Transformers (Vaswani et al., 2017), we use multi-head self-attention module as our GCN aggregation (pooling) function. It calculates the importance of individual message coming from its neighbors to generate the new aggregated output.

3.1 Reading Order Equivariant Positional Encoding (ROPE)

Positional encoding (Gehring et al., 2017) in sequence models is with an assumption that the input is perfectly serialized. However, as illustrated in Figure 1, form-like documents often contain multiple columns or sections. A simple left-to-right and top-to-bottom serialization commonly provided by OCR engines does not provide accurate sequential presentation of words – two consecutive words in the same sentence might have drastically different reading order indexes by naive serialization.

Instead of assigning absolute reading order indexes for the entire document at the beginning, we propose to encode the relative reading order context of neighboring words w.r.t. the target word based on the given graph connectivity. Figure 3 demonstrates the process of the proposed method: ROPE iterates through the neighboring word vertices in the original reading order and assigns new ROPE codes $p \in \mathbb{N}$ (red numbers) to the neighbors, starting from zero. The generated codes are then appended to the corresponding incoming messages during graph message passing. Hence, ROPE provides a relative reading order context of the neighborhood for order-aware self-attention pooling.

Note that the generated ROPE codes remain unchanged if the neighbors and the target shift equally in the document with the same relative order, therefore being equivariant. Additionally, ROPE provides robust sequential output that is consistent even when the neighborhood crosses multiple columns or sections in a document.

Finally, we also explore sinusoidal encoding matrix (Vaswani et al., 2017) besides the index-based encoding. Our ablation study in Section 4 shows that using both results in the best performance.

4 Experiments

We evaluate how reading order impacts overall performance of graph-based information extraction from form-like documents. We adopt two form understanding tasks as Jaume et al. (2019), including word labeling and word grouping. Word labeling is the task of assigning each word a label from a set of predefined entity categories, realized by node classification. Word grouping is the task of aggregating words that belong to the same entity, realized by edge classification. These two fundamental entity extraction tasks do not rely on perfect entity word groupings provided by the dataset and therefore help decouple the modeling capability provided by the proposed ROPE in practice. These two tasks

also effectively demonstrate the quality of the node embedding and edge embedding of the proposed graph architecture and decouple any performance gain from sophisticated Conditional Random Field (CRF) decoders often used on top of the model.

4.1 Datasets

Payment. We follow Majumder et al. (2020) to prepare a large-scale payment document collection that consists of around 18K single-page payments. The data come from different vendors with different layout templates. For both word labeling and word grouping experiments, we use a 80-20 split of the corpus as the training and test sets.

We use a public OCR service[1] to extract words from the payment documents. The service generates the text of each word with their corresponding 2D bounding box. The word boxes are roughly arranged in an order from left to right and from top to bottom. We then ask human annotators to label the words with 13 semantic entities. Each entity ground truth is described by an entity type and a list of words generated by the OCR engine, resulting in over 3M word-level annotations. Labelers are instructed to label all instances of a field in a document, therefore our GCNs are trained to predict all instances of a field as well.

FUNSD. FUNSD (Jaume et al., 2019) is a public dataset for form understanding in noisy scanned documents, containing a collection of research, marketing, and advertising documents that vary widely in their structure and appearance. The dataset consists of 199 annotated forms with 9,707 entities and 31,485 word-level annotations for 4 entity types: header, question, answer, and other. For both word labeling and word grouping experiments, we use the official 75-25 split for the training and test sets.

4.2 Experimental Setup

All GCN variants used in the experiment have the same architecture: The node update function is a 2-layer Multi-Layer Perceptron (MLP) with 128 hidden nodes. The aggregation function uses a 3-layer multi-head self-attention pooling with 4 heads and 32 as the head size. The number of hops in the GCN is set to 7 for payment dataset and 2 for FUNSD dataset due to the complexity and scale of the former. We use cross-entropy loss for both multi-class word labeling and binary word grouping tasks. We train the models from scratch using Adam optimizer with the batch size of 1. The learning rate is set to 0.0001 with warm-up proportion of 0.01. The training is conducted on 8 Tesla P100 GPUs for approximately 1 day on the largest corpus.

[1] cloud.google.com/vision

	Types of Positional Encoding (ours)		Word Labeling	Word Grouping		
	EdgeGeo	ROPE	F1	P	R	F1
Payment			60.80	83.64	83.97	83.80
	✓		66.09	84.96	84.93	84.94
		✓	68.17	84.92	**86.86**	85.88
	✓	✓	**74.55**	**86.75**	86.53	**86.64**
FUNSD			50.86	82.09	92.21	86.86
	✓		53.16	87.56	87.17	87.37
		✓	51.78	**88.90**	89.67	89.28
	✓	✓	**57.22**	88.64	**90.03**	**89.33**

Table 1: Different positional encodings for GCNs on information extraction tasks. We observe that the reading order encoding (ROPE) is equally or more important compared to edge geometric feature (EdgeGeo).

	ROPE Encoding Function		Word Labeling	Word Grouping		
	Index	Sinusoidal	F1	P	R	F1
Payment			66.09	84.96	84.93	84.94
	✓		72.41	**87.78**	85.31	86.53
		✓	70.94	88.49	83.00	85.66
	✓	✓	**74.55**	86.75	**86.53**	**86.64**
FUNSD			53.16	87.56	87.17	87.37
	✓		55.48	85.95	**92.15**	88.94
		✓	54.14	**88.72**	89.51	89.12
	✓	✓	**57.22**	88.64	90.03	**89.33**

Table 2: Ablation of positional encoding function used in the proposed ROPE. We observe that either index or sine encoding works better than no positional encoding. Combined works the best.

4.3 Results

We train the GCNs from scratch on all datasets. For word labeling we use multi-class node classification F1-scores as the metric and for word grouping we use binary edge classification F1-scores as the metric with the corresponding precision and recall values.

Importance of reading order. Positional encoding mechanisms are the key components to exploiting layout patterns of words – `Answer` entities are usually next to or below the `Question` entities. Existing GCN approaches rely on edge geometric (EdgeGeo) features to capture such spatial relationships between words in 2D space. Here we evaluate the importance of the proposed reading order encoding ROPE with various combinations of EdgeGeo over the baseline GCN (Qian et al.,

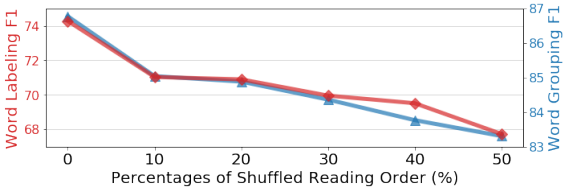


Figure 4: Sensitivity of ROPE to OCR reading order on Payment. The proposed ROPE codes remain the same if the connected neighboring words and target word shift equally in the document.

2019) as summarized in Table 1. Without any positional encoding, word labeling F1 drops by 13.75 points and word grouping F1 drops by 2.84 points on payment dataset. Then, we pass ROPE to incoming messages and find that this reduces the drop to 6.38 points on word labeling and 0.76 points on word grouping. Similar trend can be observed on FUNSD as well. Surprisingly, ROPE reduces performance drop more effectively than EdgeGeo on the larger payment dataset. Given these ablations, we conclude that reading order information is at least the same or more important than geometric features, and they bring orthogonal improvements to the overall performance.

Reading order encoding function. In practice, each target word usually has less than 8 neighboring words given a constructed β-skeleton graph. Therefore, a natural approach to assigning relative reading orders is to simply use the ROPE encoded indexes. In Table 2 we observe that simple index encoding immediately improves GCN without ROPE by 6.32 points on word labeling and 1.59 points on word grouping using payment corpus. Next we explore the popular sinusoidal function (with 3 base frequencies) for reading order encoding. It improves GCN without ROPE by 4.85 points on word labeling and 0.72 points on word grouping. Interestingly, sine function provides on par performance but does not outperform index encoding. The reason might be because the β-skeleton graph does not generate an extremely large number of neighbors, so simple index encoding is sufficient.

Sensitivity to OCR reading order. We investigate the robustness of ROPE to the quality of the input reading order. We shuffle the reading order provided by the OCR engine with a varying percentage of words before feeding into ROPE. Figure 4 exhibits the performance. For both word labeling and word grouping tasks, ROPE provides performance improvement up to less than 30% word order shuffling on the large payment corpus. With 30% or more word order shuffled, we observe less performance degradation on the word labeling, suggesting that the word grouping task is more sensitive to the original OCR reading order.

5 Conclusion

We present a simple and intuitive reading order encoding method ROPE that is equivariant to relative reading order shifting. It embeds the effective positional encoding from sequence models while leveraging the existing spatial layout modeling capability of graphs. We foresee the proposed ROPE can be immediately applicable to other document understanding tasks.

Acknowledgements. We are grateful to Evan Huang, Lauro Beltrão Costa, Yang Xu, Sandeep Tata, and Navneet Potti for the helpful feedback on this work.

References

Milan Aggarwal, Hiresh Gupta, Mausoom Sarkar, and Balaji Krishnamurthy. 2020. Form2seq: A framework for higher-order form structure extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Brian Davis, Bryan Morse, Scott Cohen, Brian Price, and Chris Tensmeyer. 2019. Deep visual template-free form parsing. In *International Conference on Document Analysis and Recognition (ICDAR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems (NIPS)*.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. 2020. Se (3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in neural information processing systems (NeurIPS)*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning (ICML)*.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. 2019. Searching for mobilenetv3. In *Proceedings of the International Conference on Computer Vision (CVPR)*.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *ICDAR-OST*.

Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards understanding 2d documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

David G Kirkpatrick and John D Radke. 1985. A framework for computational morphology. In *Machine Intelligence and Pattern Recognition*.

Frank Lebourgeois, Zbigniew Bublinski, and Hubert Emptoz. 1992. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *International Conference on Pattern Recognition (ICPR)*.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2020. Learning to encode position for transformer with continuous dynamical model. In *International Conference on Machine Learning (ICML)*.

Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. 2020. Representation learning for information extraction from form-like documents. In *proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Rasmus Berg Palm, Ole Winther, and Florian Laws. 2017. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In *International Conference on Document Analysis and Recognition (ICDAR)*.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *International Conference on Machine Learning (ICML)*.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. GraphIE: A graph-based framework for information extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems (NIPS)*.

Sandeep Tata, Navneet Potti, James B. Wendt, Lauro Beltrão Costa, Mark Najork, and Beliz Gunel. 2021. Glean: Structured extractions from templatic documents. In *International Conference on Very Large Data Bases (VLDB)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Renshen Wang, Yasuhisa Fujii, and Ashok C. Popat. 2021. Post-ocr paragraph recognition by graph convolutional networks. *arXiv preprint arXiv:2101.12741*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. In *International Conference on Machine Learning (ICML)*.

Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. 2020. Pick: processing key information extraction from documents using improved graph learning-convolutional networks. In *International Conference on Pattern Recognition (ICPR)*.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. 2020. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations (ICLR)*.

Xiaohui Zhao, Endi Niu, Zhuo Wu, and Xiaoguang Wang. 2019. Cutie: Learning to understand documents with convolutional universal text information extractor. In *International Conference on Document Analysis and Recognition (ICDAR)*.

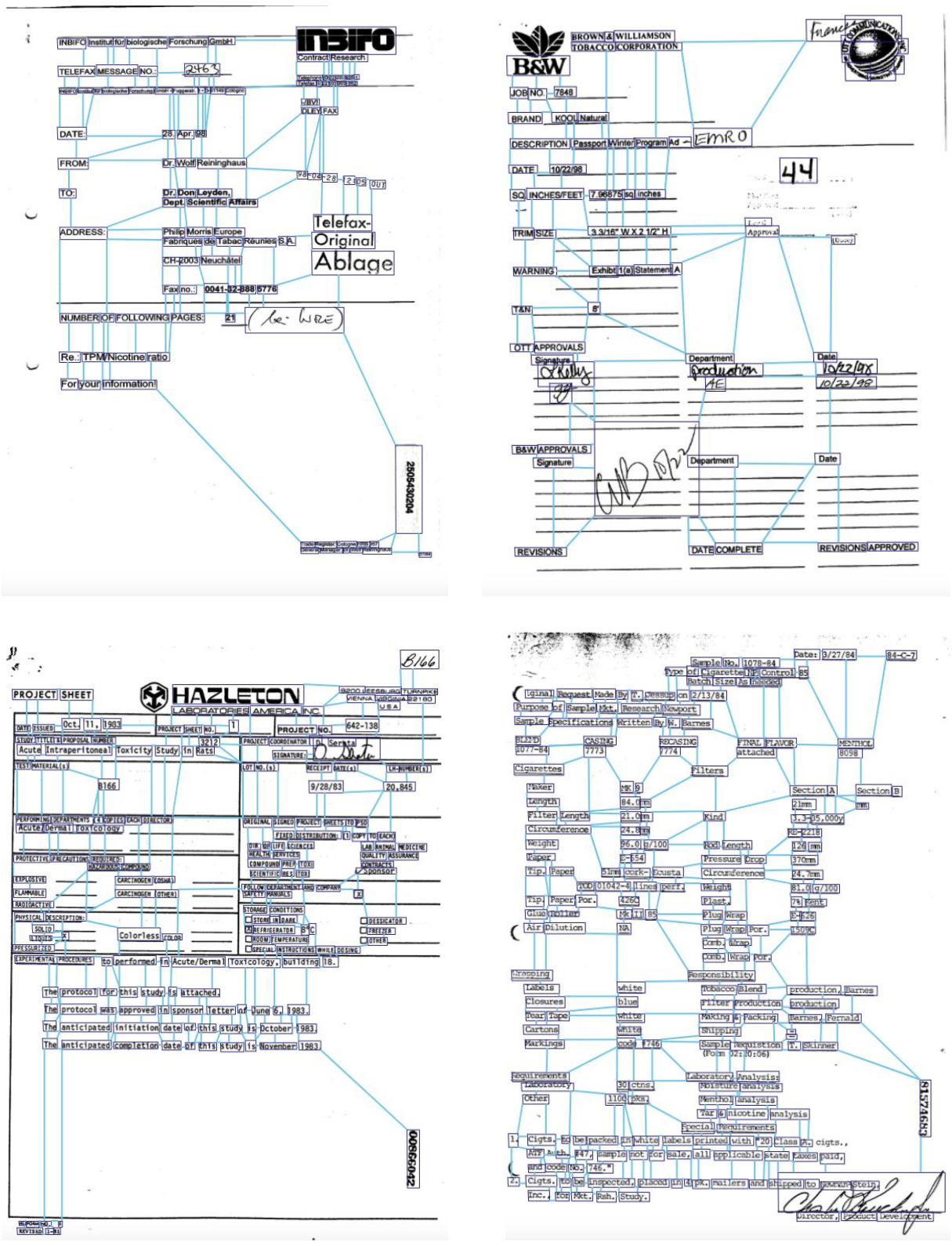


Figure 5: β-skeleton examples of documents of FUNSD. By using the "ball-of-sight" strategy, β-skeleton graph offers high connectivity between word vertices for necessary message passing while being much sparser than fully-connected graphs for efficient forward and backward computations

Figure 6: Sample output of the word grouping task on FUNSD with a few failure cases.

Zero-shot Event Extraction via Transfer Learning: Challenges and Insights

Qing Lyu[1], **Hongming Zhang**[2*], **Elior Sulem**[1], **Dan Roth**[1]

[1]Department of Computer and Information Science, UPenn
[2]Department of Computer Science and Engineering, HKUST
{lyuqing, eliors, danroth}@seas.upenn.edu
hzhangal@cse.ust.hk

Abstract

Event extraction has long been a challenging task, addressed mostly with supervised methods that require expensive annotation and are not extensible to new event ontologies. In this work, we explore the possibility of zero-shot event extraction by formulating it as a set of Textual Entailment (TE) and/or Question Answering (QA) queries (e.g. "*A city was attacked*" entails "*There is an attack*"), exploiting pretrained TE/QA models for direct transfer. On ACE-2005 and ERE, our system achieves acceptable results, yet there is still a large gap from supervised approaches, showing that current QA and TE technologies fail in transferring to a different domain. To investigate the reasons behind the gap, we analyze the remaining key challenges, their respective impact, and possible improvement directions[1].

1 Introduction

Event extraction (EE) has long been an important and challenging NLP task. Figure 1 exemplifies a TRANSFER-OWNERSHIP event from the ACE-2005 dataset (Walker et al., 2006), where the *trigger* is "*purchased*" and the *arguments* include "*China*" (Buyer), "*Russia*" (Seller), etc. The subtasks of EE involve identifying and classifying event triggers and their corresponding arguments.

The predominant approaches normally require supervision (e.g. Lin et al., 2020), which is both expensive and inflexible when moving to new event ontologies. Recent works (Chen et al., 2020; Du and Cardie, 2020) have pointed out the connection between Question Answering (QA) and EE in developing supervised systems. Meanwhile, several efforts have explored unsupervised methods. Peng et al. (2016) first attempted to extract event *triggers* with minimal supervision using similarity-based heuristics. Huang et al. (2018) and Lai et al. (2020) explored both trigger and argument extraction under a slightly different setting: training on some event types and testing on unseen ones. Recently, Liu et al. (2020) proposed a QA-based zero-shot argument extraction method, which did not handle triggers. So far, no method has been proposed to extract *both* event triggers and arguments without any EE training data[2]. Moreover, the performance of existing zero-shot attempts, especially on arguments, is still far from satisfactory, yet little is known about possible underlying reasons.

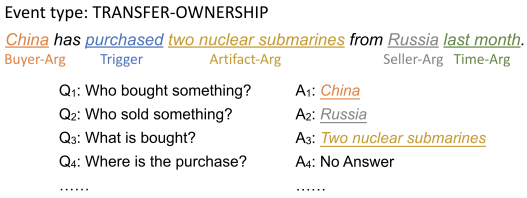


Figure 1: An example of an event from ACE-2005, and how arguments are extracted via QA.

In this work, we investigate the possibility of zero-shot EE via transfer learning from Textual Entailment (TE) and QA. Observe that given pretrained TE/QA models, extracting events can be viewed as answering questions/verifying hypotheses about a text. For example, the sentence in Figure 1, taken as the premise, would entail the hypothesis "*There is a transfer of ownership*", therefore providing the event type. Then, by asking Q_1 "*Who bought something?*", we obtain "*China*" as the Buyer. Similarly, Q_2, Q_3 will yield the Seller and Artifact, and so on.

Based on the observation above, we propose an intuitive zero-shot EE approach. It does not require any event training data, but we still make several design choices based on the development set. To demonstrate the level of generalization, we choose the optimal model with the ACE development set, and evaluate it on both ACE and ERE (LDC2015E29) test sets. The performance

[*] This work was done when the author was visiting the University of Pennsylvania.

[1]Our code and models will be available at http://cogcomp.org/page/publication_view/943.

[2]An exception is Zhang et al. (2021), done concurrently.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 322–332
August 1–6, 2021. ©2021 Association for Computational Linguistics

surpasses previous zero-shot approaches on every subtask when the gold trigger span is given, yet is still unsatisfying compared to supervised methods, revealing a large gap in using off-the-shelf TE/QA models for direct transfer. To shed light on why it is the case, we identify the key challenges behind the gap, and attribute each of them to the intrinsic weakness of pretrained models, our usage of them, or the task itself. We then anatomize their individual impact with an ablation study.

Our contributions are: (1) We propose the first TE/QA-based event extraction system that tackles *both* triggers and arguments without any event training data; (2) We show that existing TE/QA models do not support direct domain transfer well; and (3) We provide insights into the remaining challenges, their individual influence, and possible directions for future research.

2 Approach

Our pipeline consists of two modules, trigger extraction and argument extraction, both relying on pretrained TE/QA models for direct transfer.

The pretrained models we use are all BERT-based (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020), including a TE model trained on MNLI (Williams et al., 2018), a Yes/No QA model trained on BoolQ (Clark et al., 2019), and an extractive QA model trained on QAMR (Michael et al., 2018) and/or SQuAD2.0[3] (Rajpurkar et al., 2018)[4]. The TE model, when given a *premise* and a *hypothesis*, predicts the relation between them ("entailment", "contradiction", or "neutral"). The Yes/No QA model takes as input a *context* and a *Yes/No question*, and returns either Yes or No. Finally, the extractive QA model is also given a *context* but with a *Wh-question*, and the answer is a span in the context. With these models, we design the two modules for event extraction.

2.1 Trigger Extraction (T-Ext)

We formulate Trigger Extraction (T-Ext) as a TE or a Yes/No QA task. Only the TE case is illustrated, since the other only differs in the query format.

To obtain potential event triggers from a sentence, we first run Semantic Role Labeling (SRL) as a preprocessing step. We use a BERT-based Verb+Nominal SRL model[5]. The sentence is then chunked into "text pieces", each containing an SRL predicate and its core arguments (e.g. A_0, A_1, A_2).

Then, for each text piece, we pass it to the TE model as the premise, coupled with a *hypothesis* in the format of "*This text is about ...*" for each event type, inspired by Yin et al. (2019). For example, the hypothesis for BE-BORN is "*This text is about someone's birth.*". Then, for each hypothesis, the model returns the probability that it is entailed by the premise. If the highest entailment probability across all event types surpasses a threshold, we output the corresponding SRL predicate as an event trigger of this type.[6]

Argument	Question
Artifact	"What is bought?"
Buyer	"Who buys something?"
Seller	"Who sells something?"
Price	"How much does something cost?"
Beneficiary	"Who is something bought for?"
Time	"When is the purchase?"
Place	"Where is the purchase?"

Table 1: The predefined question for each argument type in an TRANSFER-OWNERSHIP event.

2.2 Argument Extraction (A-Ext)

We formalize the task of Argument Extraction (A-Ext) as a sequence of QA interactions with the pretrained extractive QA model.

Given an input sentence and the extracted trigger, we ask a set of questions based on the event type definition, and retrieve the QA model's answers as argument predictions.

Consider the example in Figure 1. Assume that T-Ext has identified a TRANSFER-OWNERSHIP event with the trigger "*purchased*". With this information, we consult a predefined set of questions for each argument type in the current event type. For instance, Table 1 provides a full collection of questions for all arguments in TRANSFER-OWNERSHIP. Finally, to obtain the head of the argument (e.g. "submarines" in "two nuclear submarines"), we implement a simple heuristics-based head identifier based on the AllenNLP Dependency Parser[7] as a post-processing step.

An important caveat in the above process concerns missing arguments. Specifically, many argument types in the event template do not occur in every sentence, e.g. in Figure 1, there is no Place argument. For simplicity, we call questions with a non-empty gold answer "has-answer" (HA) ques-

[3] Abbreviated as SQuAD henceforth.

[4] See Appendix A and B for model and dataset details.

[5] https://github.com/CogComp/SRL-English

[6] See Appendix C.2 for configuration details.

[7] https://demo.allennlp.org/dependency-parsing

Setting	System	TI	TI+TC	AI	AI+AC
scratch (supervised)	Lin et al. 20	78.2	74.7	59.2	56.8
scratch (zero-shot)	Huang et al. 18[8]	55.6	49.1	**27.8**	15.8
	Zhang et al. 20	**58.3**	**53.5**	16.3	6.3
	Ours	45.5	41.7	27.0	**16.8**
gold TI (zero-shot)	Huang et al. 18	-	33.5	-	14.7
	Zhang et al. 20	-	82.9	-	-
	Ours	-	**83.7**	**38.9**	**24.2**
gold TI+TC (zero-shot)	Liu et al. 20	-	-	-	25.8
	Ours	-	-	**44.3**	**27.4**

Table 2: The F1 score on ACE-2005. Subtasks include Trigger Identification (TI), Trigger Classification (TC), Argument Identification (AI), and Argument Classification (AC). See Section 3 for setting definitions. SOTA results among zero-shot methods are in boldface.

tions and the rest "no-answer" (NA) questions. The QA model is considered to output NA when it predicts an empty span or the highest non-empty span confidence is lower than a threshold.

3 Experimental Setup

We evaluate our system on the ACE-2005 dataset. Its event ontology has 7 types and 33 subtypes, and we evaluate T-Ext directly on the subtypes. The same train/development/test split from Lin et al. (2020) is used. We make several design choices[9] on the development and report results on the test, ignoring the training set.

To demonstrate how our model generalizes, we also directly evaluate the optimal model on the ERE dataset (LDC2015E29). To adapt to ERE, we define a query for each new event type.

There are four subtasks of event extraction: Trigger Identification (**TI**), Trigger Classification (**TC**), Argument Identification (**AI**), and Argument Classification (**AC**). We experiment under three settings: **scratch**, where the system performs all subtasks without any gold annotation; **gold TI**, where gold trigger spans are given; **gold TC**, where gold trigger spans and types are given[10].

Following Ji and Grishman (2008), Precision, Recall, and F1 are used for evaluation[11]. We evaluate argument spans on the head level, consistent with most prior work (Huang et al., 2018; Wadden et al., 2019; Lin et al., 2020; Zhang et al., 2021).

4 Results

We report results in comparison with several existing zero-shot methods (Huang et al., 2018; Liu et al., 2020; Zhang et al., 2021), as well as a supervised SOTA system (Lin et al., 2020).

Setting	System	TI	TI+TC	AI	AI+AC
scratch (supervised)	Lin et al. 20	68.4	57.0	50.1	46.5
scratch	Ours	39.8	31.8	23.0	15.0
gold TI		-	58.4	30.8	18.8
gold TI+TC (zero-shot)		-	-	47.9	27.5

Table 3: The F1 score on the ERE. The optimal model is chosen on ACE dev and directly evaluated on ERE.

As shown in Table 2, on the ACE test set, our system outperforms prior zero-shot methods in every subtask under both the "gold TI" and "gold TI+TC" settings. However, it fails in "scratch", indicating that the main bottleneck lies in identifying exact trigger spans. Compared with the supervised SOTA, our system is still notably worse on TI, AI, and AC in particular, like other zero-shot systems.

Table 3 shows the results on ERE. Compared to ACE, our argument detection module generalizes well, whereas the trigger module does not. Under the gold TI setting, the TC F1 on overlapping event types is 70.4, whereas on new event types it is only 19.0, likely because the newly added event types in ERE have a finer definition. For example, a model needs to understand "whether a contact is in-person or not" to distinguish between MEET (in-person), CORRESPONDENCE (not in-person), and CONTACT (unsure). Further research should focus on how to effectively generalize to new event types with subtle definitions.

5 Analysis

Using the results on ACE, we now present an analysis of the remaining core challenges of the task, along with an ablation study on their individual impact. To further understand the challenges, we attribute each to the fragility of the pretrained *models* (**M-Error**), our *usage* of the models (**U-Error**), or the *task* itself (**T-Error**).

5.1 Trigger Extraction

5.1.1 Error Analysis

We first analyze the distribution of error types. Specifically, we manually check 100 wrong predictions and show the counts in Figure 2(a). Only the most frequent types are discussed here, and the remaining can be found in Appendix E.1.1.

Subtle trigger (M-Error): This is the main intrinsic error from the TE model (17%). Event types like DIE & EXECUTE, ATTACK & INJURE, and MEET & PHONE-WRITE are especially confus-

[8] Trained on 10 event types; tested on unseen ones.

[9] See Appendix C.2 and C.3.

[10] We don't have a **gold AI** setting, since the proposed QA-based A-Ext module cannot do AC alone.

[11] Evaluation scripts are adapted from `http://blender.cs.illinois.edu/software/oneie`.

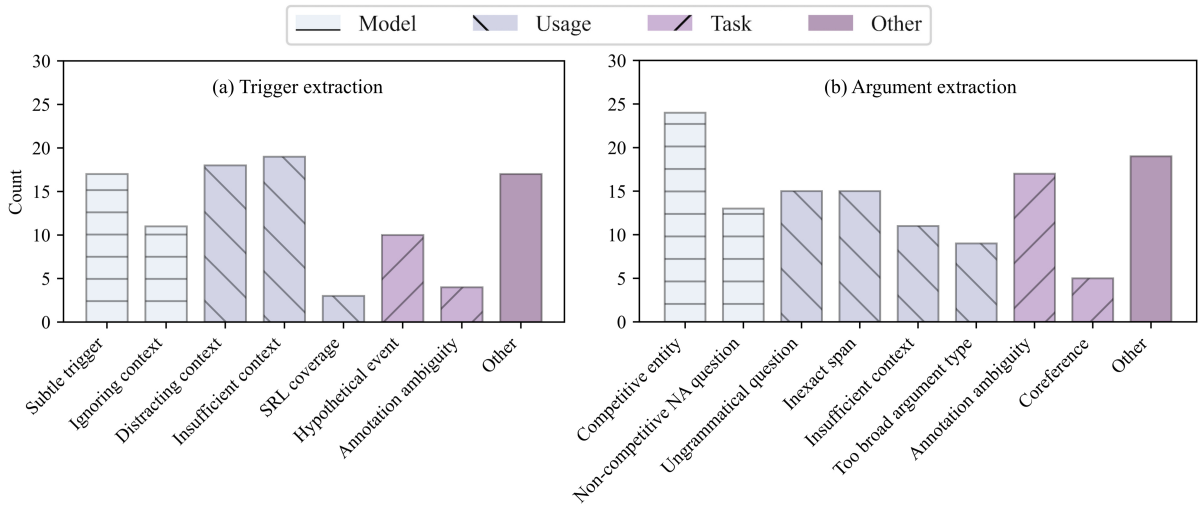


Figure 2: Error types in trigger and argument extraction in 100 wrong predictions. The count sum exceeds 100 since a prediction can contain multiple types of error. Colors/patterns indicate the origin of the error type.

ing. Though their definitions slightly differ, the model fails to capture this level of subtlety.

Distracting & Insufficient context (U-Error): Two other error types from our usage of the TE model concern distracting (18%) or insufficient (19%) contexts. An example of distracting context is "*The woman's parents ... found the decomposing body*". Given the word "decomposing", the model predicts it as a DIE event trigger, due to "*body*" in the premise. In contrast, insufficient contexts provide too little information. For example, in the sentence "*Turkey sent 1,000 troops ... and said it would send more*", the TE model is asked to predict the event type of "*send*" but only sees "*it send more*" as the premise, since "troops" is not part of the SRL arguments of "*send*". As a result, the model predicts a TRANSFER-MONEY instead of TRANSPORT event.

Hypothetical event & Annotation ambiguity (T-Error): Finally, two error types stem from the task itself: "hypothetical events" (10%) and "annotation ambiguity" (4%) . Hypothetical events refer to sentences like "*They will not buy it if it is too expensive*", where the TE model predicts "*buy*" as a TRANSFER-OWNERSHIP event trigger. Though such events should be annotated as per the ACE Annotation Guideline (3.4), this is not always strictly followed. Other cases of inconsistent annotation also cause errors, e.g. among all occurrences of "*give birth to*", the trigger is "*give*" in some cases, while "*birth*" in others.

5.1.2 Ablation Study

We further explore the two U-Error types, by measuring their influence on the performance while controlling for other factors. Only one type is included in this section, and the remaining can be found in Appendix E.1.2.

Premise design: To see the impact of **insufficient & distracting context**, we select all instances of these two types, and change the premise design. The re-prediction is done under gold-TI. For insufficient contexts, the premise is now the entire sentence. For distracting contexts, we adopt a "minimal-pair premise" strategy: Premise A is the original (e.g. "*...decomposing body...*"); Premise B is formed by deleting the candidate trigger from A (e.g. "*...body...*"). Then, we take the event type with the highest entailment probability *difference* between A and B as the prediction. Intuitively, this difference signifies the semantic *contribution* of the candidate trigger toward an event type.

After re-prediction, 59% errors are corrected on insufficient contexts. Among the remaining 41%, it is either the case that the model still ignores the context, or that the longer context now brings distraction.

On distracting contexts, only 18% errors are corrected. The model still cannot overcome the distraction in most remaining errors, which suggests that a more complicated strategy is needed in addition to manipulating the premise.

5.2 Argument Extraction

5.2.1 Error Analysis

Likewise, we analyze 100 wrong argument predictions and discuss several major error types. Figure 2(b) shows their respective counts. For a full explanation, see Appendix E.2.1.

Competitive entity & Non-competitive NA ques-

tions (M-Error): The QA model is intrinsically weak on "competitive entities" (24%) and "non-competitive NA questions" (13%).

When identifying an argument for the target event, another entity of the same type, i.e. a "competitive entity", can co-occur in the context. For example, the sentence "*A unit ... meets in confidential sessions to review terrorist activities in Europe*" has a MEET event. When asked "*Where is the meeting*", our model answers "*Europe*" whereas the gold answer is empty, since "*in Europe*" is attached to "*activities*". We find that models trained on extractive QA data are easily fooled by such entities, if they are of the desired type asked by the question. Note that competitive entities can occur for both HA and NA questions.

The other type involves NA questions without any competitive entity. For example, given the sentence "*Iraqi forces responded with artillery fire*", the question "*When is the fire*" has no answer, and there is no Time-type entity to distract the model. However, the model can still give arbitrary answers (e.g. "*artillery*") with very high confidence, due to its inherent incapacity for NA questions.

Ungrammatical question (U-Error): This relatively frequent error type (15%) is attributable to our usage of the QA model. To facilitate the model to better locate the target event, we embed the trigger in the questions whenever possible, which sometimes unavoidably makes them ungrammatical. For example, our question for the Place argument in a TRANSFER-OWNERSHIP event is "*Where is the* {*trigger*}". This is only grammatical when the trigger is a noun. Thus, the QA model may be confused by such questions.

5.2.2 Ablation Study

To isolate A-Ext, we perform the ablation study under the gold TI+TC setting. We explore four error types involving both M-Error and U-Error, two of which are included in this section, the rest in Appendix E.2.2.

Pretraining data: To examine the influence of **NA questions**, we compare QA models trained on QAMR (He et al., 2020) and SQuAD2.0, only the latter of which has NA questions. Results show that the one trained on QAMR greatly outperforms the one on SQuAD (+16.9 on AI; +13.6 on AC). To unveil why it is the case, we propose three hypotheses: (1) QAMR and ACE both have one-sentence contexts, while SQuAD has paragraphs. (2) The NA questions in SQuAD "confuses" the model, i.e. SQuAD and ACE have similar types of HA questions, while different types of NA questions. (3) The *density* of answers per sentence is high in both QAMR and ACE, while low in SQuAD. We test each hypothesis using controlled experiments, but none of them turns out to provide a full explanation of the performance difference[12].

Moreover, we train a binary classifier for HA and NA questions on a balanced sample of SQuAD, resulting in over 86 in-domain accuracy. On ACE, this number drops to 57. This shows that the QA model cannot even distinguish well between HA and NA questions when it comes to a new dataset, let alone answer them.

Question grammaticality: To see the impact of **ungrammatical questions**, we manually correct the grammatical error and re-predict with the model. Among all relevant wrong predictions, 40% are now correct. The rest 60% are mostly also NA questions that prove to require more than just fixing the grammar to solve.

6 Conclusions

We propose the first complete zero-shot event extraction system via transfer learning from TE and QA. While QA/TE models perform exceptionally well on standard benchmarks (SQuAD, QAMR, MNLI), they do not generalize as expected when being used on EE datasets. We analyze the limited success and several main challenges of the current approach, and provide insights for future improvements.

Acknowledgments

This work was supported in part by Contracts FA8750-19-2-1004 and FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA) and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contracts No. 2019-19051600006 and 2019-19051600004 under the BETTER Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

We thank Celine Lee and Hangfeng He for providing the SRL and QAMR models respectively. We also thank Ying Lin, Jian Liu, Lifu Huang, Haochen Zhang, and the anonymous reviewers for their valuable help and/or feedback.

[12] Details can be found in Appendix E.2.2.

References

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. Quase: Question-answer driven sentence encoding. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-Shot Transfer Learning for Event Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: Hlt*, pages 254–262.

Viet Dac Lai, Thien Huu Nguyen, and Frank Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event Extraction as Machine Reading Comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 560–568, New Orleans, Louisiana. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event Detection and Co-reference with Minimal Supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Unsupervised label-aware event trigger and argument classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL) Findings*.

A Dataset Statistics

The pretraining datasets we use include MNLI (Williams et al., 2018), BoolQ (Clark et al., 2019), QAMR (Michael et al., 2018), and SQuAD2.0 (Rajpurkar et al., 2018). Our evaluation dataset is ACE-2005 (LDC2006T06) and ERE (LDC2015E29). Table 4 shows the number of examples in each dataset.

Dataset	Train	Dev	Test
MNLI	392,702	20,000	20,000
BoolQ	9,427	3,270	3,245
QAMR	73,561	27,535	26,994
SQuAD2.0	130,319	11,873	8,862
ACE-2005	17,172	923	832
ERE	-	-	2,069

Table 4: Number of examples in all datasets used.

B Details on Pretrained Models

We use three different pretrained representations, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and BART (Lewis et al., 2020). All models are implemented with HuggingFace Transformers[13].

The pretrained model checkpoints we use include: `bert-base-uncased` (110M parameters), `bert-large-uncased` (336M parameters), `roberta-base` (125M parameters), `roberta-large` (335M parameters), `facebook/bart-base` (373M parameters), `facebook/bart-large` (406M parameters)[14].

For TE and Yes/No QA, we finetune the pretrained models using the standard `SequenceClassification` pipeline. For extractive QA, we finetune the models using the `QuestionAnswering` pipeline[15]. The finetuning scripts are adapted from the `text-classification` and `question-answering` examples in the HuggingFace Transformers repository[16]. The hyperparameter values and pretrained models will be made available via the HuggingFace model sharing service.

We run our experiments on an NVIDIA GeForce RTX 2080 Ti GPU, with half-precision floating point format (FP16) with O1 optimization. The finetuning take 3 hours to 20 hours depending on the task.

C Details on Event Extraction System

We include here a full list of hyperparameter configurations explored in building our event extraction system. To select the optimal configuration, we perform grid-search on the development set based on the F1 score.

C.1 Preprocessing

We adapt the preprocessing script from Lin et al. (2020)[17]. In addition, we use several general-purpose NLP tools to further process the text, including a Part-of-Speech Tagger, a Dependency Parser, a Constituency Parser[18].

C.2 Trigger Extraction Module

Pretrained representation As said in Appendix B, we experiment with three representations (BERT, RoBERTa, and BART) with their base and large versions.

[13] https://github.com/huggingface/transformers

[14] All models above are available at https://huggingface.co/transformers/pretrained_models.html

[15] Both pipelines are available from https://huggingface.co/transformers/model_doc/

[16] https://github.com/huggingface/transformers/tree/master/examples/legacy

[17] http://blender.cs.illinois.edu/software/oneie

[18] The POS tagger is from http://www.nltk.org/; the rest are from https://demo.allennlp.org/.

Pretraining task We have two pretraining task choices, TE (using MNLI as training data) and Yes/No QA (using BoolQ as training data).
SRL constituents in the premise For each predicate, we only include itself and a few core arguments to form the premise. The combinations we try include: Predicate only; Predicate, Arg0, Arg1, Arg2; Predicate and all arguments.
Confidence threshold For an SRL predicate to be identified as an event trigger, we require that the confidence score of the TE model on the "Entailment" label (resp. the Yes/No QA model on the "Yes" label) exceeds a threshold. We search the threshold value within the range of $[0.80, 0.85, 0.90, 0.95, 0.99]$.
Hypothesis format We experiment with two strategies to phrase the hypothesis:

- *Topical*: The hypothesis is in the format of "*This text is about* {*topic*}", where the "{*topic*}" is predefined for each event type. For example, for ATTACK, the hypothesis is "This text is about an attack".

- *Natural*: The hypothesis is in a natural language format. For example, for ATTACK, it is "*Someone is attacked*"[19].

The optimal configuration for trigger extraction is:
- Pretrained representation: RoBERTa-large;
- Pretraining task: TE;
- SRL arguments in the premise: Predicate, Arg0, Arg1, Arg2;
- Confidence threshold: 0.99;
- Hypothesis format: Topical.

C.3 Argument Extraction Module

Pretrained representation As said in Appendix B, we experiment with three representations (BERT, RoBERTa, and BART) with their base and large versions.
Pretraining data We have two extractive QA datasets for pretraining, SQuAD2.0 and QAMR (and also their combination).
Question format We experiment with two question formats:

- *Static*: The questions are fixed for each event type. For example, the question for the Place argument in an ATTACK event is always "*Where is the attack?*".

- *Contextualized*: The questions are instantiated with the trigger of event instances when possible. For example, the question for the Place argument in an ATTACK event is "*Where is the* {*trigger*}*?*", where "{*trigger*}" is the specific trigger token(s) of the current event instance[20].

Confidence threshold For the extractive QA model to predict a non-empty answer, we require that its confidence score should be higher than a threshold. We search within the range of $[0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99]$.

The optimal configuration for argument extraction is:
- Pretrained representation: RoBERTa-large;
- Pretraining data: QAMR;
- Question format: Contextualized;
- Confidence threshold: 0.0 (the threshold value makes almost no difference, since most model prediction confidence scores are over 0.99).

D Full Results

Complementary to Section 4, Table 5 and Table 6 shows the full results including Precision, Recall, and F1 score on ACE and ERE respectively.

E Analysis (Continued)

This section elaborates on the remaining error types and ablation study experiments not covered by Section 5.

E.1 Trigger Extraction

E.1.1 Error Analysis

Ignoring context (M-Error): This is another prevalent error type (11%), which can also be attributed to the TE model. The model focuses too much on the candidate trigger itself while disregarding the context. Consider the sentence "*He was instrumental in creating such shows as 'married with children'...*". The word "*married*" is wrongly predicted as a MARRY event trigger. The TE model identifies it as an actual event rather than the name of a show.
SRL coverage (U-Error): Among all errors, 3% originate from the fact that the target trigger is not covered by SRL in the first place. This is a matter

[19] See the Supplemental Material for a list of all hypotheses.

[20] See the Supplemental Material for a list of all questions.

[21] Trained on 10 event types; tested on unseen ones.

Setting	System	TI			TI+TC			AI			AI+AC		
		P	R	F	P	R	F	P	R	F	P	R	F
scratch (supervised)	(Lin et al. 2020)	-	-	78.2	-	-	74.7	-	-	59.2	-	-	56.8
scratch (zero-shot)	(Huang et al. 2018)[21]	85.7	41.2	55.6	75.5	36.3	49.1	28.2	27.3	**27.8**	16.1	15.6	15.8
	(Zhang et al. 2020)	58.9	57.8	**58.3**	54.6	53.5	**54.0**	19.8	38.9	26.3	9.4	18.5	12.5
	Ours	34.7	66.3	45.5	31.7	60.6	41.7	20.2	40.4	27.0	12.6	25.2	**16.8**
gold TI (zero-shot)	(Huang et al. 2018)	-	-	-	-	-	33.5	-	-	-	-	-	14.7
	(Zhang et al. 2020)	-	-	-	-	-	82.9	-	-	-	-	-	-
	Ours	-	-	-	-	-	**83.7**	35.1	43.7	**38.9**	21.8	27.2	**24.2**
gold TI+TC (zero-shot)	(Liu et al. 2020)	-	-	-	-	-	-	-	-	-	25.5	26.0	25.8
	Ours	-	-	-	-	-	-	39.4	50.7	**44.3**	24.4	31.4	**27.4**

Table 5: The full performance on ACE-2005.

Setting	System	TI			TI+TC			AI			AI+AC		
		P	R	F	P	R	F	P	R	F	P	R	F
scratch (supervised)	(Lin et al. 2020)	-	-	68.4	-	-	57.0	-	-	50.1	-	-	46.5
scratch	Ours	34.5	68.2	45.8	30.2	59.7	40.1	18.2	37.9	25.1	12.1	24.3	16.1
gold TI		-	-	-	-	-	80.0	33.6	41.1	37.0	21.0	25.7	23.1
gold TI+TC (zero-shot)		-	-	-	-	-	-	39.4	50.6	44.3	24.4	31.3	27.4

Table 6: The full performance on ERE.

of our usage of the TE model. Specifically, current SRL systems cannot handle nominal triggers perfectly, and cannot detect multi-word triggers like "*step aside*" or adjectival triggers like "*dead*" at all.

Others: Other less-frequent error types besides those mentioned in the main text are related to coreference (e.g. when pronouns like "*this*" are triggers,), proper names (e.g. historical events like "*intifada*"), confidence scores being too low (thus not identifying a gold trigger), ambiguity of the hypothesis (e.g. a "*nuclear test*" is predicted as a TRIAL-HEARING event because of the word "*test*" and the hypothesis "*There is a trial or hearing*".)

E.1.2 Ablation Study

SRL models: To examine the influence of **SRL coverage**, we experiment with two more SRL models: Illinois SRL (Punyakanok et al., 2008)[22], and one that identifies almost every verb and nominal[23]. None of the three can identify adjectival/multi-word predicates. In comparison, every model can cover over 90% verb triggers, while the nominal trigger coverage varies from 60% to 95%. On T-Ext, the highest-coverage model performs the best (+4.0 F1 on TI, +6.8 on TC over the lowest-coverage model), proving that the gain from greedy identification does compensate for the cost in precision.

Pretraining task: Our results show that the TE-based TC far outperforms its Yes/No QA counterpart (by 52.6%). One hypothesis is that the pretraining data for the TE model (MNLI; about 400K examples) is much larger than that for the QA model (BoolQ; about 9K). To verify that, we retrain a TE model on a portion of MNLI of the same size as BoolQ. As a result, the gap shrinks to 31.4%, though still quite large. This proves the importance of the training data size. It also implies that in order to further improve the current TE-based method, using larger-scale training data might be promising.

Hypothesis design: It is observed that the hypothesis format also plays a nontrivial role. As said in Appendix C.2, we experiment with two hypothesis designs, *topical* and *natural*. Experiments show that "topical" is better than "natural" by 1.9% on TC, suggesting the sensitivity of current TE systems to the phrasing of texts.

E.2 Argument Extraction

E.2.1 Error Analysis

Too broad argument type (M-Error/U-Error): For this error type (9%), both the model and our usage are to blame. Though ACE has a strict definition of arguments, the QA model sometimes interprets them too broadly. For instance, with the context "*A blindfolded woman was shot in the head by a hooded militant*", given the question "*Where is the shot*", the model answers "*in the head*". This is not technically wrong, but certainly not the desired Place argument either. We cannot hold the QA model entirely accountable, since the questions

[22] https://cogcomp.seas.upenn.edu/page/software_view/SRL

[23] Also from https://github.com/CogComp/SRL-English.

are indeed too generic as well.

Inexact span (U-Error): 15% errors are because of the inexact match of gold and predicted argument spans. For instance, the gold is "*Saturday morning*" while the predicted is "*morning*". Though in our evaluation, we compare only heads of the phrases whenever possible, not all ACE arguments (i.e. those of the "value" type instead of the "entity" type) have head annotations. Under this circumstance, the current evaluation framework does not give credit to a partial match, which can be an imperfection for potential improvement.

Insufficient context (U-Error): Like in trigger extraction, the model is sometimes given insufficient context when predicting arguments (11%). The target argument can be entirely outside the SRL constituents of the predicate, thus making it impossible to extract.

Coreference & Annotation ambiguity (T-Error): Error types ascribed to the task include "coreference" (5%) and "annotation ambiguity" (17%). The former refers to the case when the model predicts a coreferent of the gold argument. However, the current evaluation framework still takes it as an error. The latter happens when the model makes a sensible prediction, yet it is inconsistent with the annotation. For example, in the sentence "*Iraqi forces responded with artillery fire*", the model recognizes "*artillery*" as the Instrument for the ATTACK event triggered by "fire". However, no Instrument is annotated. Future evaluation framework should consider allowing multiple correct answers in such cases of human disagreement.

Others: Other errors are related to multiple arguments (i.e. the model only predicts one of them), lacking document-level knowledge (i.e. the sentence itself is not informative enough), and also arbitrary predictions with no obvious reason.

E.2.2 Ablation Study

Pretraining data: Continuing from the "Pretraining data" paragraph in Section 5.2.2, we test three hypotheses for the gap between training on QAMR and SQuAD.

Hypothesis(1): QAMR and ACE both have one-sentence contexts, while SQuAD has paragraphs.

We try to verify it by retraining a QA model on a new version of QAMR with longer contexts, subject to the same length distribution of SQuAD. This is done by either a) adding random sentences, or b) repeating the original sentence. It is observed that a) almost doesn't hurt AI at all but AC a little (3%), and b) lowers AI by 4% and AC by 3%. Therefore, though longer contexts do weaken the performance slightly, it is not the main reason behind the gap between QAMR and SQUAD.

Hypothesis(2): The NA questions in SQuAD "confuses" the model, i.e. SQuAD and ACE have similar types of HA questions, while different types of NA questions.

To test this hypothesis, we retain all HA questions in SQuAD to make a new dataset. We also construct a control set of the same size, but with both NA and HA questions randomly sampled from the original SQuAD. We retrain a QA model on each dataset, and find that the HA-only set brings about an increment by 7% on AI but a drop by 2% on AC, compared to the control set. This suggests that the addition of NA questions in SQuAD does have mixed effects on event extraction. Future research should focus on how to better transfer a model's ability to identify NA questions to a different domain.

Hypothesis (3): The *density* of answers per sentence is high in both QAMR and ACE, while low in SQuAD.

To see if this is the cause, we construct a new version of QAMR by retaining only one QA pair for each sentence. A control set of the same size, but with multiple QA pairs per sentence, is also constructed by randomly deleting sentences (along with all their QA pairs) from the original QAMR. Results show that the low-density set is only worse than the control set on AI by 0.5% and on AC by 0.2%, indicating that the density of answers is not a critical aspect.

Type constraints in question: Since generic questions may have been a cause for **too broad argument types**, we experiment with a new set of question templates that contain specific entity-type requirements whenever possible. For example, instead of "*Where is the shot*", we ask "*What is the location of the shot*", which may prevent the model from answering "*in the head*". However, only 11% errors are fixed after re-prediction, indicating that encoding type constraints is non-superficial.

Question design: Like the hypothesis format in trigger extraction, the design of questions also makes a difference for arguments. As mentioned in Appendix C.3, we explore two formats, *static* and *Contextualized*. Experiments show that switching from "static" to "contextualized" boosts AI by 7%

while impairs AC by 3%, suggesting that contextualized questions overall helps the model better locate the event.

Context design: To measure the influence of *insufficient context*, we now use the entire sentence as the context on these instances, similar to trigger extraction. Results show that 27% of them are now correct, and another 27% are partially correct (inexact span).

Using Adversarial Attacks to Reveal the Statistical Bias in Machine Reading Comprehension Models

Jieyu Lin[2], **Jiajie Zou**[2], **Nai Ding**[1,2]*
[1]Zhejiang Lab / Hangzhou, China
[2]Key Laboratory for Biomedical Engineering of Ministry of Education, College of Biomedical Engineering and Instrument Sciences, Zhejiang University / Hangzhou, China
{ljy5905,jiajiezou,ding_nai}@zju.edu.cn

Abstract

Pre-trained language models have achieved human-level performance on many Machine Reading Comprehension (MRC) tasks, but it remains unclear whether these models truly understand language or answer questions by exploiting statistical biases in datasets. Here, we demonstrate a simple yet effective method to attack MRC models and reveal the statistical biases in these models. We apply the method to the RACE dataset, for which the answer to each MRC question is selected from 4 options. It is found that several pre-trained language models, including BERT, ALBERT, and RoBERTa, show consistent preference to some options, even when these options are irrelevant to the question. When interfered by these irrelevant options, the performance of MRC models can be reduced from human-level performance to the chance-level performance. Human readers, however, are not clearly affected by these irrelevant options. Finally, we propose an augmented training method that can greatly reduce models' statistical biases.

1 Introduction

Reading comprehension tasks are useful to quantify language ability of both humans and machines (Richardson et al., 2013; Xie et al., 2018; Berzak et al., 2020). Deep neural network (DNN) models have achieved high performance on many MRC tasks, but these models are not easily explainable (Devlin et al., 2019; Brown et al., 2020). It is also shown that DNN models are often sensitive to adversarial attacks (Jia and Liang, 2017; Ribeiro et al., 2018; Si et al., 2019, 2020). Furthermore, it has been shown DNN models can solve MRC tasks with relatively high accuracy when crucial information is removed so that the tasks are no longer solvable by humans (Gururangan et al., 2018; Si et al., 2019; Berzak et al., 2020). All such evidence suggests that the high accuracy DNN models achieve on MRC tasks does not solely rely on these models' language comprehension ability. At least to some extent, the high accuracy reflects exploitation of statistical biases in the datasets (Gururangan et al., 2018; Si et al., 2019; Berzak et al., 2020).

Here, we propose a new model-independent method to evaluate to what extent models solve MRC tasks by exploiting statistical biases in the dataset. As a case study, we only focus on the classic RACE dataset (Lai et al., 2017), which requires MRC models to answer multiple-choice reading comprehension questions based on a passage. The advantage of multiple-choice questions is that its performance can be objectively evaluated. At the same time, it does not require the answer to be within the passage, allowing to test, e.g., the summarization or inference ability of models. Nevertheless, since models are trained to select the right option from 4 options, which are designed by humans and may contain statistical biases, models may learn statistical properties of the right option. Consequently, models may tend to select options with these statistical properties similar to the properties of the right option without referring to the passage and question. Our method is designed to reveal this kind of statistical bias.

The logic of our method is straightforward: For each multiple-choice question, we gather a large number of options that are irrelevant to the question and passage. We ask the model to score how likely each irrelevant option is the right option. If a model is biased, it may always assign higher scores to some irrelevant options than others, even if all the options are irrelevant. If a model is so severely biased, which turns out to be true for all models tested here, it may assign higher scores to some irrelevant options than the true answer and select the irrelevant option as the answer. Here, the irrelevant

*Corresponding author: Nai Ding

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 333–342
August 1–6, 2021. ©2021 Association for Computational Linguistics

options that are often selected as the answer are referred to as magnet options.

2 Dataset and Pre-trained Models

We used RACE dataset in our experiment (Lai et al., 2017), which is a large-scale reading comprehension data set covering more than 28,000 passages and nearly 100,000 questions. The task was to answer multi-choice questions based on a passage. Specifically, each question contained a triplet (p_i, q_i, o_i), where p_i denoted a passage, q_i denoted a question, and o_i denoted a candidate set of 4 options, i.e., $o_i = \{o_{i,1}, o_{i,2}, o_{i,3}, o_{i,4}\}$. Only one option was the correct answer, and the accuracy was evaluated by the percent of questions being correctly answered.

We tested 3 pre-trained language models, i.e., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019). For each model, we separately tested the base version and large version. We built our models based on pre-trained transformer models in the Huggingface (Wolf et al., 2020). We fine-tuned pre-trained models based on the RACE dataset and the parameters we used for fine-tuning were shown in Appendix A.1.

The passage, question, and an option were concatenated as the input to models, i.e., [CLS, p_i, SEP, q_i, $o_{i,j}$, SEP]. The 4 options were separately encoded. The concatenated sequence was encoded through the models and the output embedding of CLS was denoted as $C_{i,j}$. We used a linear transformation to convert vector $C_{i,j}$ into a scalar $S(o_{i,j})$, i.e., $S(o_{i,j}) = WC_{i,j}$. The scalar $S(o_{i,j})$ was referred to as the score of the option $o_{i,j}$. A score was calculated for each option, and the answer to a question was determined as the option with the highest score, i.e., $\text{argmax}_j S(o_{i,j})$.

3 Adversarial Method

3.1 Screen Potential Magnet Options

We evaluated potential statistical biases in a model by giving it a large number of irrelevant options. For each question, we augmented the options using a set of irrelevant options, i.e., $O_A = \{o_{a1}, o_{a2}, ..., o_{aN}\}$. O_A was randomly selected from the RACE dataset with 2 constraints. First, the options belonged to questions that were not targeted at passage p_i. Second, none of the options in O_A was identical to any of the original options in o_i. The augmented question was denoted as $(p_i, q_i, \{o_{i,1}, o_{i,2}, o_{i,3}, o_{i,4}, o_{a1}, ..., o_{aj}, ..., o_{aN}\})$. A score was independently computed for each option using the procedure mentioned above. Since the options in O_A were irrelevant, an ideal model should never select them as answers. If $\text{max}_j S(o_{i,j}) < S(o_{ak})$ for any k, however, the model would select the k^{th} irrelevant option as the answer. We define an interference score T_k using the following equation.

$$T_k = \frac{1}{N}\sum_{i=1}^{N} T_{i,k}, \qquad where$$

$$T_{i,k} = \begin{cases} 1, & if \quad \text{max}_j S(o_{i,j}) < S(o_{ak}) \\ 0, & otherwise \end{cases}$$

For an ideal model, $T_{i,k}$ should always be 0. For a model that makes mistakes but shows no consistent bias, the interference score should be comparable for all o_{ak}. If the model is biased, the interference score may be always high for some options so that the model always selects them as the answer whether they are relevant to the question or not.

3.2 Adversarial Attack

We constructed an adversary attack to the MRC models using one magnet option. For each question, we replaced a wrong option with a magnet option, i.e., o_{ak}. The replaced option set was $\{o_{i,1}, o_{i,2}, o_{i,3}, o_{ak}\}$. The passage and the question were not modified, and the answer did not change. An example was shown in Figure 1. If the model chooses the original answer even when a magnet option is introduced, it is stable, not sensitive to the attack. In contrast, if it chooses the magnet option, i.e., o_{ak}, as the answer, it is successfully attacked.

4 Results and Analyses

4.1 Experiments Setup

To screen potential magnet options, we constructed a large set of irrelevant options, i.e., O_A, by randomly selecting 300 passages from the RACE test set, which were associated with 1064 questions. Furthermore, to test whether options in the training set can cause stronger interference, we also randomly selected 300 passages from the RACE training set, which had 1029 questions. The options from the test and training set were pooled to create O_A, which had 8372 options in total.

Passage: "...Quantum computers could be able to do what modern supercomputers are unable to do by using transistors that are able to take on many states at the same time..."

Question: According to the text, quantum computing _ .

Original Options:	**Adversarial Options:**
A. can reduce the cost of computers	*A*. can reduce the cost of computers
B. can make computers run by themselves	*B*. misfortune may be an actual blessing
C. **will work by using transistors**	*C*. **will work by using transistors**
D. has been put in use so far	*D*. has been put in use so far
Model Choice: *C* – correct	**Model Choice:** *B* – incorrect, **successfully attacked**
A, B, or D – incorrect	*C* – correct, not attacked
	A or D – incorrect, not attacked

Figure 1: An example of the task and adversarial attack. The option in bold is the true answer, and the option in red indicates the irrelevant option that was used for attack.

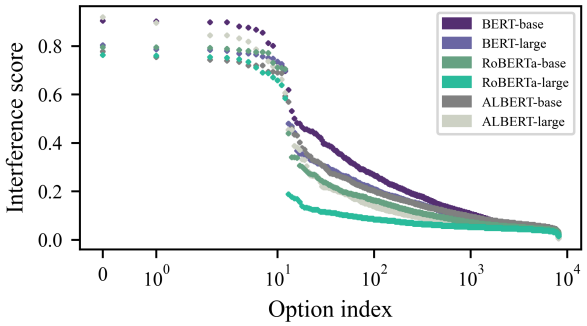


Figure 2: Interference score evaluated based on a subset of questions.

For such a large number of irrelevant options, it was computationally challenging to evaluate the interference score of each option based on each question in the RACE test set. Therefore, as a screening procedure, we first randomly selected 100 passages from the RACE test set, which have a total of 346 questions. The interference score for each of the 8372 irrelevant options was evaluated based on the 346 questions.

After potential magnet options were determined by the screening procedure, the interference score of magnet options were further evaluated using all questions in RACE test set. For RACE test set, the accuracy of the models ranged between about 0.6 and 0.85, with RoBERTa-large achieving the highest performance (Table 1).

4.2 Screening for Magnet Options

The interference score for 8372 options was independently calculated for each model. Results were shown in Figure 2, where the interference score was sorted for each model. It is found that most of the irrelevant options had a non-zero interference score, and some irrelevant options yielded high interference scores around 0.8, which meant the models would choose those irrelevant options as the answer for about 80% of the questions. Irrelevant options from the training and test sets had similar interference scores (Appendix B.1).

It was found that the options with exceptionally high interference scores around 0.8 were options that combined other options, such as "all the above", which were called the option-combination series. However, not all the magnet options were from the option-combination series. Normal statements, e.g., "The passage doesn't tell us the end of the story of the movie", could also reach an average interference score around 0.34.

The correlation between the interference score between models were shown in Appendix B.2. We separately showed the results for options from the option-combination series and the others. The correlation coefficient between models had an average value around 0.76, which proved that the interference score was correlated across models. From another perspective, it also implied that our method could work as a model-insensitive adversarial attack method.

4.3 Validate Magnet Options and Adversarial Attack

We further evaluated the interference score of potential magnet options based on all the questions in the RACE test set. To construct a set of magnet options for this analysis, we averaged the interference score across 3 models, i.e., BERT-large, RoBERTa-large, and ALBERT-large. All options in O_A were sorted based on the average score, and we selected 20 options with the highest interference scores to construct the magnet option set, with the

	BERT		ALBERT		RoBERTa	
Version	base	large	base	large	base	large
Original accuracy	0.614	0.681	0.683	0.752	0.738	0.846
Adversarial accuracy[1]	0.094	0.167	0.217	0.064	0.166	0.297
Adversarial accuracy[2]	0.381	0.524	0.334	0.506	0.656	0.798

Table 1: Model performance on the RACE test set and model performance after being attacked. The superscript 1 meant use "A, B and C" to attack, and the superscript 2 meant use "The passage doesn't tell us the end of the story of the movie" to attack.

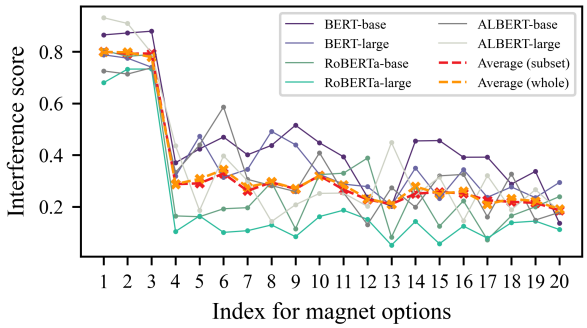


Figure 3: Interference score evaluated based on the whole RACE test set.

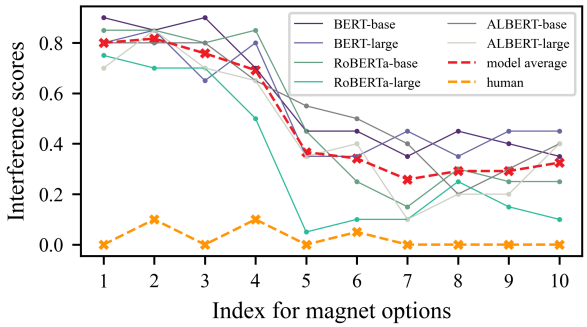


Figure 4: Interference score for the human experiment and the corresponding interference scores for the models.

following constraint: Since options with the highest interference scores were often from the option-combination series, to increase diversity, we only included 3 options from the option-combination series. We listed all the 20 magnet options in Appendix A.2. The interference score calculated based on the whole RACE test set was shown in Figure 3, which was very similar to the results based on the subset of 346 questions in Figure 2 (comparing average-whole and average-subset in Figure 3).

Table 1 showed the accuracy of models when attacked by 2 example magnet options. When attacked, the model performance could drop by as much as 0.68.

4.4 Human Evaluation

Next, we verified whether humans were also confused by the magnet options. We randomly selected 20 questions and 10 magnet options. The 10 magnet options selected were listed in Appendix A.3. Ten questions were not modified while the other 10 questions were attacked using the procedure shown in Figure 1. Twenty human evaluators answered these 20 questions online. The accuracy of humans did not reduce under attack (0.90 in the original samples and 0.94 in the adversarial samples). The interference score for humans, also the corresponding interference score for the models, was shown in Figure 4. Humans were not confused by the magnet options.

4.5 Training with Adversarial Examples

To reduce sensitivity to magnet options and to potentially reduce the statistical biases of MRC models, we proposed an augmented training method and tested the method using the base version of all models. In the augmented training method, 400 options with the highest interference scores were selected as the irrelevant option set. For each question in the RACE training set, the option set was augmented by adding an option randomly chosen from the irrelevant option set. In other words, although each original question has 4 options, during the augmented training each question has 5 options, including the 4 original options and a randomly chosen irrelevant option. We fine-tuned pre-trained models based on the training set with augmented options.

The accuracy of models fine-tuned using augmented options were shown in Table 2, comparable to the original accuracy in Table 1. When attacked, however, the accuracy of models fine-tuned using augmented options were much higher than the adversarial accuracy in Table 1.

The 1000 options with the highest interference

base version	BERT	ALBERT	RoBERTa
Original accuracy	0.601	0.689	0.723
Adversarial accuracy[1]	0.576	0.681	0.725
Adversarial accuracy[2]	0.670	0.740	0.778

Table 2: Model performance on the RACE test set based on augmented training.

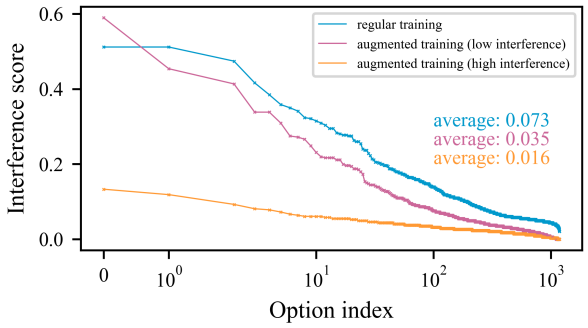


Figure 5: Interference score of 1186 randomly chosen options that are not used in augmented training.

scores were selected to evaluate the effect of augmented training, as shown in Appendix C. Result showed that the interference score dropped for both the 400 options used for augmented training and the other 600 options that were not used for training. Therefore, the effect of augmented training could generalize to samples not used for augmented training.

Another experiment was implemented to explore the impact of irrelevant option set selection. We separately used options with high and low interference scores for training and found that options with higher interference score were more effective at reducing statistical biases (Figure 5).

4.6 Interference Score Analysis

Did the statistical biases revealed in previous analyses originate from the pre-training process or the fine-tuning process? Without fine-tuning, the pre-trained models performed poorly on RACE. However, results showed that such an imprecise model could show strong biases (Appendix B.3). Interestingly, the interference score was not correlated between the pre-trained model and the fine-tuned model, suggesting that fine-tuning overrode the biases caused by pre-training and introduced new forms of biases.

5 Related Work

Our attack strategy distinguishes from previous work in two ways. First, unlike, e.g., gradient-based methods (Ebrahimi et al., 2018; Cheng et al., 2020), our method does not require any knowledge about the structure of DNN models. Second, some methods manipulate the passage in a passage-dependent way (Jia and Liang, 2017; Si et al., 2020; Zhao et al., 2018), while our method manipulate the options in a passage-independent way. Furthermore, we proposed a strategy to train more robust models that are insensitive to our attack.

Here, we restricted our discussion to RACE, but our method is applicable to other tasks in which the answer is selected from a limited set of options. For example, for span extraction tasks, such as SQuAD, the method will insert a large number of irrelevant phrases into the passage and analyze which phrases are often selected as the answer. In this way, our method is similar to the trigger-based attack methods (Wallace et al., 2019), but the difference is that our method test whether the inserted irrelevant phrase is selected as the answer while the trigger-based methods test whether the content following the trigger phrase is selected.

6 Conclusion

In summary, we propose a new method to evaluate the statistical biases in MRC models. It is found that current MRC models have strong statistical biases, and are therefore sensitive to adversarial attack. When attacked using the method proposed here, model performance can drop from human-level performance to chance-level performance. To alleviate sensitivity to such attacks, we provided an augmented training procedure that effectively enhances the robustness of models.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful suggestions and comments. Work supported by Major Scientific Research Project of Zhejiang Lab 2019KB0AC02 and National Natural Science Foundation of China 31771248.

References

Yevgeni Berzak, Jonathan Malmaud, and Roger Levy. 2020. STARC: Structured annotations for reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5726–5735, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3601–3608. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Qiu Ran, Peng Li, Weiwei Hu, and Jie Zhou. 2019. Option comparison network for multiple-choice reading comprehension. *arXiv preprint arXiv:1903.03033*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does BERT learn from multiple-choice reading comprehension datasets? *CoRR*, abs/1910.12391.

Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. 2020. Benchmarking robustness of machine reading comprehension models. *CoRR*, abs/2004.14004.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen,

Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2018. Large-scale cloze test dataset created by teachers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2344–2356, Brussels, Belgium. Association for Computational Linguistics.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2020. DCMN+: dual co-matching network for multi-choice reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9563–9570. AAAI Press.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

	BERT		ALBERT		RoBERTa	
version	base	large	base	large	base	large
learning rate	1.00E-05	1.00E-05	2.00E-05	1.00E-05	1.00E-05	1.00E-05
train epochs	5	5	/	/	4	4
train steps	/	/	12000	12000	/	/
train batch size	16	24	32	32	16	16
warmup steps	0	0	1000	1000	1200	1200
weight decay	0	0	0	0	0.1	0.1

Table 3: Hyperparameters for fine-tuning on RACE. We adapted these hyperparamemers from Lan et al. (2019); Liu et al. (2019); Ran et al. (2019); Zhang et al. (2020).

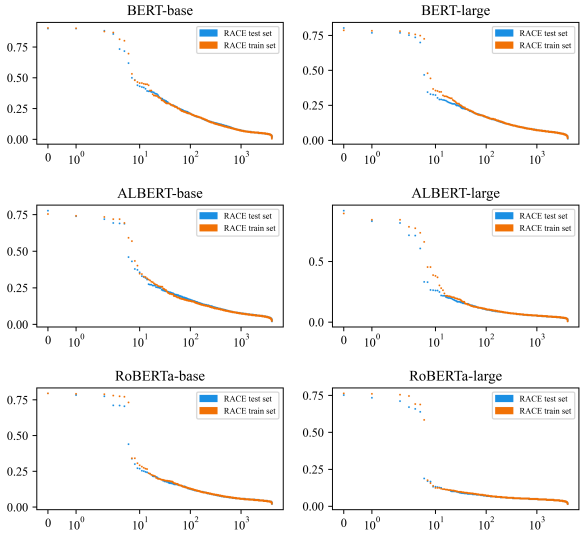


Figure 6: Interference score evaluated based on a subset of questions.

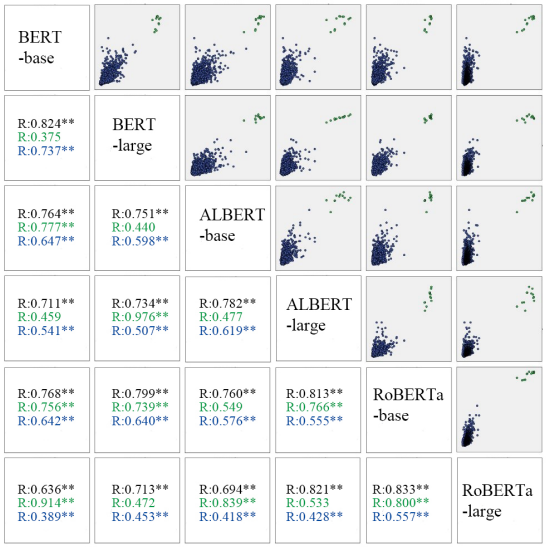


Figure 7: The scatter matrix diagram of the interference scores of the irrelevant options among models.

A Experimental Details

A.1 Fine-tuning Parameters

The parameters we used in the process of fine-tuning the pre-trained models were shown in Table 3.

A.2 Magnet Options for Validate

The 20 magnet options used for evaluating the interference scores in Section 4.3 were shown as following. The sentences selected from the RACE training set were shown in bold.

1. A, B and C
2. **all of A, B and C**
3. All of the above.
4. **Not all of it can be avoided.**
5. It's well beyond what the author could be responsible for.
6. **The passage doesn't tell us the end of the story of the movie**
7. didn't give the real answer
8. **make us know it's important to listen to people who offer a different perspective through his experience**
9. **give us a turning point in mind**
10. **not strictly stuck to**
11. You should purposely go out and make these mistakes so that you can learn from them and not have them ruin your entire life.
12. what's inside a person is much more important than his/her appearance.
13. **Not all of it is man-made Ming dynasty structure.**
14. **introduce the topic of the passage**
15. **The central command didn't exactly state what had caused the crash.**
16. **one good turn deserves another.**
17. the growing population is not the real cause of the environment problem.,
18. **misfortune may be an actual blessing.**

BERT-base	Correlation coefficient	accuracy	Average interference score
Pre-trained model	-0.023	0.315	0.0518
Partly fine-tuned model	0.069*	0.315	0.0214
Fine-tuned model	1	0.613	0.0713
RoBERTa-base	Correlation coefficient	accuracy	Average interference score
Pre-trained model	-0.021	0.225	0.3553
Partly fine-tuned model	0.088**	0.289	0.2282
Fine-tuned model	1	0.743	0.0569
ALBERT-base	Correlation coefficient	accuracy	Average interference score
Pre-trained model	-0.013	0.254	0.1483
Partly fine-tuned model	0.231**	0.39	0.1043
Fine-tuned model	1	0.702	0.0703

Table 4: Interference score of 1000 randomly selected irrelevant options for the same model architecture before and after fine-tuning. Correlation coefficient was counted between the interference score before and after fine-tuning (**$P < 0.01$, and *$P < 0.05$).

19. may meet with difficulties sometimes
20. good answers are always coming when we think outside of the box

A.3 Magnet Options for Human Evaluation

The 10 magnet options used for human evaluating in Section 4.4.

1. all the above
2. Both B and C
3. do all of the above
4. A and B
5. not strictly stuck to
6. The passage doesn't tell us the end of the story of the movie
7. It's well beyond what the author could be responsible for.
8. You should purposely go out and make these mistakes so that you can learn from them and not have them ruin your entire life.
9. make us know it's important to listen to people who offer a different perspective through his experience
10. Not all of it is man-made Ming dynasty structure.

B Study of Interference Score

B.1 Comparison of Irrelevant Options from RACE Training and Test Set

Different models in Figure 2 were separately shown in Figure 6. It denoted that irrelevant options from the training and test sets had similar interference score. Only in BERT-large and ALBERT-large models, the interference scores of the irrelevant options from the training set were higher than those

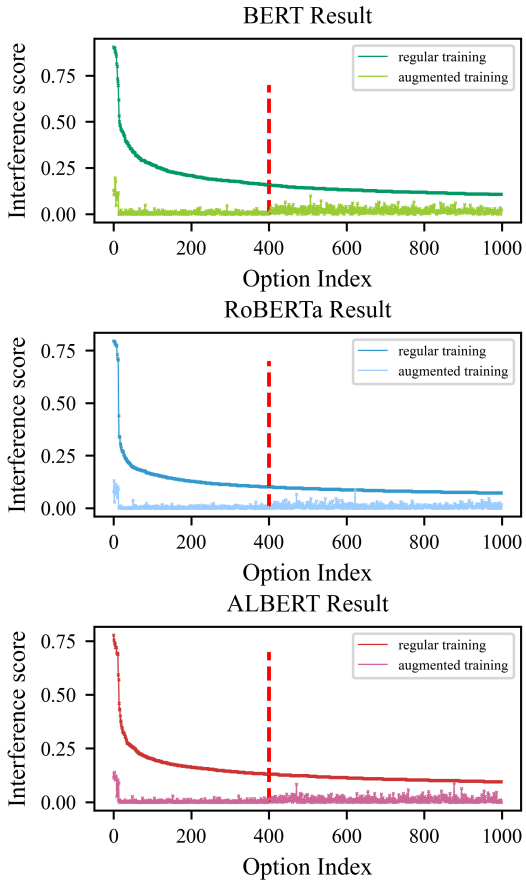


Figure 8: Interference score comparison of models evaluated based on a subset of questions.

from the test set in a certain range.

B.2 Comparison of Interference Scores Based on Different Models

The scatter matrix diagram of the interference scores of the irrelevant options among different models was shown in Figure 7. The detailed experimental process was described in Section 4.2. Here, text in black showed the correlation coefficient of all options; text in green showed the options of the option-combination series; text in blue showed the options except the option-combination series.

In general, the interference scores between models had high correlation coefficients. Models from the same architecture were more likely to have similar interference scores.

B.3 Comparison of Interference Scores During Fine-tuning

For each model architecture, the pre-trained model, partly fine-tuned model (fine-tuned the linear transformation mentioned in Section 2), and fully fine-tuned model were collected, and were used to evaluate the interference score of 1,000 randomly selected irrelevant options. The results were shown in Table 4. The subset of questions mentioned in Section 4.1 were used to evaluate the interference score.

C Augmented Training Result

The augmented training results were shown in Figure 8. In the figures, the left side of the red line contains the irrelevant options that were used in augmented training, and the right is the irrelevant options that were not involved in augmented training.

Quantifying and Avoiding Unfair Qualification Labour in Crowdsourcing

Jonathan K. Kummerfeld
Computer Science & Engineering
University of Michigan, Ann Arbor
jkummerf@umich.edu

Abstract

Extensive work has argued in favour of paying crowd workers a wage that is at least equivalent to the U.S. federal minimum wage. Meanwhile, research on collecting high quality annotations suggests using a qualification that requires workers to have previously completed a certain number of tasks. If most requesters who pay fairly require workers to have completed a large number of tasks already then workers need to complete a substantial amount of poorly paid work before they can earn a fair wage. Through analysis of worker discussions and guidance for researchers, we estimate that workers spend approximately 2.25 months of full time effort on poorly paid tasks in order to get the qualifications needed for better paid tasks. We discuss alternatives to this qualification and conduct a study of the correlation between qualifications and work quality on two NLP tasks. We find that it is possible to reduce the burden on workers while still collecting high quality data.

1 Introduction

Workers using Amazon Mechanical Turk earn a median wage of $2.54 an hour (Hara et al., 2018), far below the U.S.-federal minimum wage of $7.25. Many researchers pay workers a higher wage, estimating the time spent on a task and giving bonuses when the time required is higher than expected. At the same time, researchers try to maintain the quality of work completed using a variety of methods (Mitra et al., 2015). One common approach, used by 19% of tasks (HITs) on the platform (Hara et al., 2018), is to restrict tasks to workers who have had a certain number of HITs approved. Tasks with this restriction have a median wage of $4.14 an hour, far above the overall average. If most high paying requesters use this restriction it means workers need to do a substantial amount of low paid "Qualification Labour": work to achieve the qualifications necessary for fairly paid tasks. These tasks may also be particularly unpleasant work that more experienced workers are unwilling to do, e.g., they might involve unsavoury content.

This paper is the first to identify the qualification labour issue and explore it. We study norms around the setting of the qualification and the effort workers put in to achieve common milestones. 5,000 accepted tasks, a common requirement, takes over 2 months of effort. We consider several ways to address the issue, and study the work quality of groups with different qualifications.[1] Using two tasks, coreference resolution and sentiment analysis, we find that high quality annotations can be collected with a lower threshold, though there are task dependent patterns.

2 Background and Related Work

Crowd work involves large groups of workers doing small paid tasks, known as Human Intelligence Tasks (HITs). Services such as Amazon Mechanical Turk provide a marketplace to connect workers with requesters. Requesters create tasks, workers choose which tasks to do, then either complete them or return them. Requesters approve or reject the completed work. Tasks can be restricted to workers with certain qualifications, e.g. location. Amazon tracks some statistics that can be used as qualifications. This work focuses on (1) the total number of approved HITs a worker has, and (2) the percentage of their HITs that were accepted.

Since the earliest uses of crowd work in NLP, there has been work discussing issues such as poor wages and the lack of worker rights (Fort et al., 2011). These have also been discussed in the Human-Computer Interaction research community (Bederson and Quinn, 2011; Hara et al., 2018). There has been work on proposing guidelines for requesters (Sabou et al., 2014), incorporating workers into the IRB process (Libuše Hannah Vepřek,

[1] Code for our experiments is attached to this paper

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 343–349
August 1–6, 2021. ©2021 Association for Computational Linguistics

2020), and developing tools to help workers address the power imbalance in the online workplace (Irani and Silberman, 2013, 2016). Concurrent with this work, another study showed that crowdsourcing is being used more each year in NLP research, and there is limited awareness of the ethical issues in this type of work (Shmueli et al., 2021).

Prior work has considered hidden labour in the day-to-day work of the crowd (Hara et al., 2018). By observing a large set of workers, they measured time involved in searching for tasks, returned tasks, and breaks. Some of these issues have received additional attention, such as the wasted effort on tasks that are returned rather than completed (Han et al., 2019). While informative, those studies do not account for the hidden labour identified in this paper, which spans a long period and relates to worker qualifications.

Part of this work uses online discussion between workers to understand their work. Prior work has used a similar approach to understand the overall experience of crowd workers (Martin et al., 2014).

3 Norms for the Approved HITs Value

The value used as the Approved HITs threshold is rarely reported in prior work. Three recent papers specify a 1,000 HIT threshold (Vandenhof, 2019; Oppenlaender et al., 2020; Whiting et al., 2019). Outside of Computer Science, advice in articles (Young and Young, 2019) and tutorials (Dozo, 2020) is to set the value to 100 because that is when another qualification (approval percentage) becomes active. This difference may be because other fields primarily use crowdsourcing for surveys rather than data annotation or human computation systems. It is unclear how representative these samples are. However, there are other sources that can provide information about conventions.

One source is Amazon itself. The Mechanical Turk web-interface provides six options: 50, 100, 500, 1,000, 5,000, 10,000. The MTurk blog has mentioned this qualification in four posts over the past eight years (Amazon Mechanical Turk, 2012, 2019, 2017, 2013). In three cases, the value was 5,000 and in the fourth it was 10,000.

Another source is forums and blogs. One pinned thread on the MTurk Crowd forum advises that "For your first 1000 HITs you may want to concentrate on approval milestones rather than $$$ ... most of the better-paying requesters require 1000/5000/10000+ approved HITs" ([jklmnop], 2016). This advice is repeated elsewhere on the forum and on Reddit ([WhereIsTheWork], 2019; [CaptainSlop], 2019; [Crazybritzombie], 2018). This is consistent with observations that 80% of tasks available to new users pay less than 10 cents (El Maarry et al., 2018). In one discussion between a worker and a requester, the worker recommended a threshold of 5,000 ([clickhappier], 2016). In the blog "Tips For Requesters On Mechanical Turk", one post recommends at least 5,000 if not 10,000 (Miele, 2012) while another recommends at least 1,000 (Miele, 2018). A web article by a Computer Vision researcher recommended 1,000 (Kumar, 2014). The CloudResearch blog mentions the threshold once, noting that a value of 10,000 maintains quality without significantly increasing the time to finish a set of HITs (Robinson, 2015).

Qualifications are also discussed by courses and tutorials. In the Crowdsourcing & Human Computation course at the University of Pennsylvania, a guest lecture on "The Best Practices of the Best Requesters" mentioned the approved HITs qualification and used 10,000 as an example (Milland, 2016). One guide recommends a cutoff of 5,000 (Carlson , née Feenstra).

Overall, we conclude that while practises vary, 5,000 or higher are commonly used as a qualification for tasks.

3.1 Impact on Workers

It is difficult to estimate how much time workers have to spend to achieve this qualification. Academic studies of time spent on HITs may be skewed by experienced workers, who have strategies for finding and completing tasks rapidly. Posts on Reddit mention taking anywhere from a month to a year to reach 5,000 approved HITs. The median of values reported across several Reddit threads was 2.25 months ([alisonlovepowell], 2015; [GnomeWaiter], 2013; [FrobozzYogurt], 2020; [Wat3rloo], 2016). Assuming 20 hours of work a week that is almost 200 hours of effort (140 seconds per task).

3.2 Potential Solutions

If this type of qualification undercuts our commitment to paying a fair wage, what are alternative ways to maintain quality? Options include:

1. Introduce screening questions that workers must complete correctly to proceed to the rest of the task, e.g., requiring 70%+ on three questions (Shvartzshnaid et al., 2019). This approach is prob-

lematic because it the workers who fail the screening are doing unpaid labour.

2. Address quality after collection by either dropping the lowest performing workers (e.g., the bottom 25% in Bansal et al., 2019), aggregating a larger number of responses per example, or including attention check questions and discarding workers who get them wrong. All of these incur a substantial cost to researchers.

3. Controlled crowdsourcing (Roit et al., 2020) uses an initial task that a broad set of workers can complete and then limits participation to the workers who did well on that task.[2] The cost of this solution depends on the percentage of workers who do well on the initial task.

4. Lower the threshold, reducing the required volume of earlier work. This reduces, but does not eliminate the qualification labour issue.

These methods can also be combined. Controlled crowdsourcing (method 3) with a very low Accepted HITs threshold (method 4) for the initial task would address the ethical concern we raise here while limiting the additional cost to the recruitment phase. Attention checks and aggregation (method 2) would then address natural variation in skill and attention during large-scale annotation.

4 Studying the Approved HITs Value

All of the options above have tradeoffs that will be task dependent and in practise some combination is most likely to be the best approach. The first three have been studied in prior work, but the impact of lowering the threshold has not. In this section, we consider the quality of work completed by workers grouped by how many tasks they have previously completed and what percentage were accepted.[3]

4.1 Tasks

Coreference Resolution This is an unusual task for crowdsourcing, with a novel user interface, shown in Figure 1. Workers were shown a 244 word document from the Ontonotes dataset (Hovy et al., 2006). We identified noun phrases using the Allen NLP parser (Gardner et al., 2018) and asked workers to identify when one of two specific entities was mentioned. This is not the complete coreference resolution task, but a useful subset. We refined the task over several rounds of trial annotation to ensure the instructions were clear and the interface was efficient. Workers were asked to check their answers if they tried to submit in less than 75 seconds. If they labeled 8 items in the first 19 words, they were reminded to only label the two entities specified. We estimated that the task would take 3 minutes and paid workers 60 cents ($12 / hour). Reviews on TurkerView (https://turkerview.com/) indicated that workers effective hourly rates were $7.88, $11.25, $12.93, and $14.59.

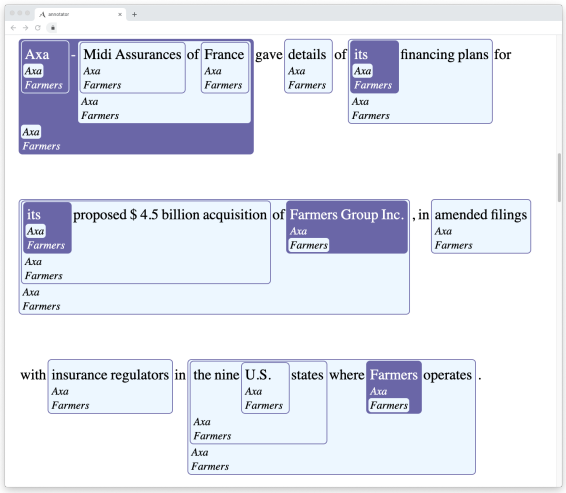


Figure 1: The user interface for coreference resolution (zoomed in). Spans are noun phrases automatically assigned by the Allen NLP syntactic parser (Gardner et al., 2018). The two entities being identified are the two most frequently mentioned entities in the text. Workers select a label by clicking on it.

We measure performance by comparing with the Ontonotes annotations. An F-score of 80% or above was considered acceptable, to allow for minor errors and points of confusion.

Sentiment Analysis This task is very intuitive and has been crowdsourced extensively in the past. We closely followed the set up used to annotate the Stanford Sentiment Treebank (Socher et al., 2013), with the same task instructions. Workers were shown ten examples whose true scores were evenly spread across 0 to 1. We estimated that the task would take 4 minutes and paid workers 80 cents ($12 / hour). Three reviews of the task on TurkerView indicated that workers hourly earnings were $22.15, $48.00, and $50.53, suggesting that workers were faster than anticipated.

[2] One potential drawback of this approach is that the filtering step may produce a biased sample of workers. That may be problematic for more subjective tasks, though with a large enough sample, responses could be weighted to make the results more representative.

[3] This was completed as part of a larger study approved by the Michigan IRB under study ID HUM00155689.

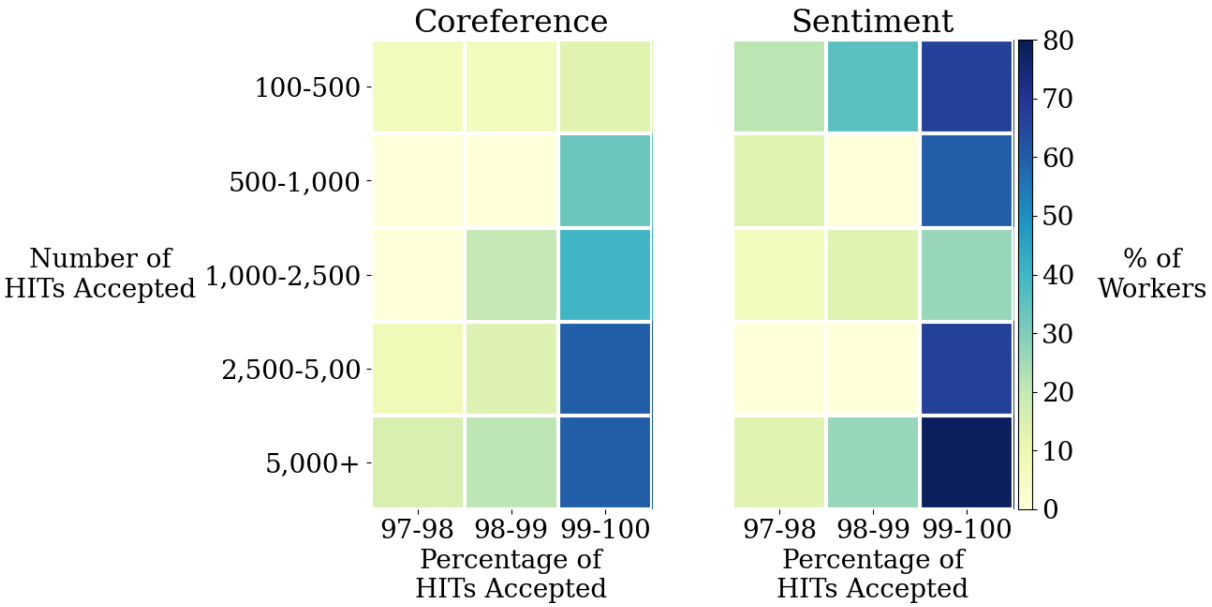


Figure 2: Results for all fifteen combinations of qualifications. Left (coreference): The percentage of workers scoring above 80 in each group. Right (sentiment): The percentage of workers whose average error was below 0.15 in each group. Each value is based on fifteen workers, except for sentiment there were fourteen for (98-99%, 500-1,000), and for coreference there were fourteen for (97-98%, 500-1,000), (97-98%, 1,000-2,500), (98-99%, 2,500-5,000), (98-99%, 5,000+), thirteen for (97-98%, 5,000+), and twelve for (97-98%, 2,500-5,000).

To evaluate, the labels are mapped to [0, 1] and compared with the STS values. An average value of below 0.15 was considered acceptable. This cutoff was chosen based on the scores achieved by two NLP students in our lab (0.11 and 0.09).

4.2 Recruitment

We considered 15 combinations of ranges for "Approved HITs" and "Percentage Approved", as shown by the axis labels in Figure 2. The ranges are based on the preset values provided by MTurk, with the addition of a boundary at 2,500 to provide slightly more detail in the shift between 1,000 and 5,000. Workers also had to be U.S.-based. We used Javascript-based checks to ensure each worker completed the task only once. 224 workers completed the sentiment task and 30 opened and returned it. 216 workers completed the coreference task and 657 opened and returned it. All but two conditions had 14 or 15 workers (the 97-98%, 5,000+ case for coreference had 13 and the 97-98%, 2,500-5,000 case for coreference had 12).

4.3 Results

The heatmap on the left of Figure 2 shows the percentage of workers scoring 80 or higher on the coreference resolution task. When the acceptance percentage is below 99, results are consistently poor, with fewer than 25% of workers scoring above 80. When the acceptance percentage is 99-100, groups with higher approved HITs have better scores. However, Figure 3 shows that more workers returned the HIT[4] in the groups with higher performance (see the last column of the rightmost plot), indicating that workers are self-selecting out.

This figure may be interpreted to suggest that a threshold of 2,500 is necessary. However, the distribution of workers is not uniform across these qualification groups. In a follow up experiment with constraints of 99-100% and 1,000+ using a relatively new requester account, 60 out of 92 workers scored 80 or above (65%), indicating that there are more workers in the higher approved HITs groups.

Figure 2 also shows results for the sentiment task. First, note that many more workers did well on the task. Comparing the left and right, the trend for percentage of HITs accepted is repeated, with consistently poor performance from workers with values below 99% (the left two columns). While the best result is the same in both cases (the bottom-right), the trend in the third column is somewhat different. Rather than a steady increase in performance as the approved HITs threshold increases, there is a U-shaped pattern. This shows that the pattern is somewhat task dependent.

[4] 'Returning' a task means a worker choose to stop working on it, receives no pay, but also receives no penalty in their profile for failing to complete the task.

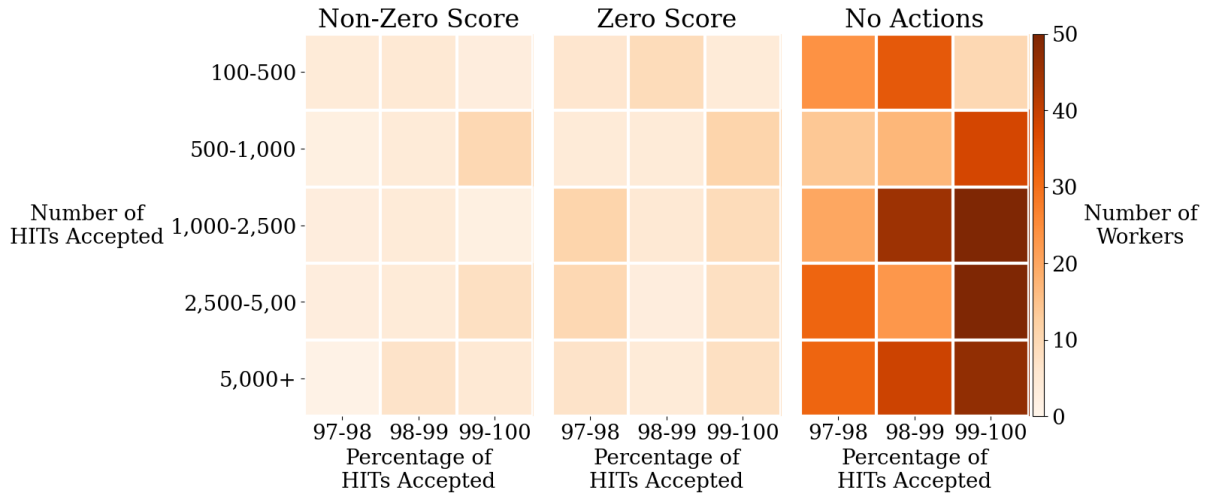


Figure 3: For coreference resolution, 657 workers opened and returned the HIT without completing it. These three heatmaps show the number of workers who: left partially correct annotations (Non-Zero Score), left entirely incorrect annotations (Zero Score), did not interact with the page (No Action). We do not include plots for sentiment analysis because only 30 workers opened and returned the HIT.

These results suggest that a lower qualification can be used without a substantial impact on work quality. In both tasks, the percentage HITs accepted qualification had a clear impact, with substantial decreases in quality from workers with a value below 99%. While that qualification does not directly force workers to do a substantial amount of work, it can be impacted by requesters who unfairly reject work. Our results also suggest that simply paying workers more will not lead to better work, as the sentiment analysis task paid considerably better and did not solve the issue.

5 Ethics and Impact Statement

This work involved consideration of several potential impacts. In terms of privacy, all data from workers is aggregated for the purpose of presenting results, and information from worker discussions were only sourced from publicly shared content. In terms of payment, we estimated the effort involved and aimed to pay workers $12 USD an hour. See the main text for worker reported values of hourly earnings on the two tasks. This was approved by the Michigan IRB under study ID HUM00155689. One potential harm of this work is that it may encourage higher values of the Percentage of HITs Accepted qualification, making workers more vulnerable to requesters who unfairly reject work.

6 Conclusion and Recommendations

This paper identifies the issue of Qualification Labour: the implied labour created by the qualifications we define. Based on a range of sources, we found that 5,000 approved tasks is one common threshold. That takes approximately two months to achieve and the tasks are poorly paid. We conducted a study of two tasks to understand how work quality correlates with these qualifications. We found that trends are task dependent, but lower thresholds can often be used.

We recommend either not using the ”HITs accepted” qualification, or running preliminary tests to identify the lowest suitable threshold for your task. This calibration is necessary because worker performance depends on many factors, including the task type, data (including which language), user interface, and instructions. One particularly promising method is to use controlled crowdsourcing (Roit et al., 2020) with a low threshold: run a short task with low or no qualifications to identify workers, then for the full task only allow those workers to participate. This reduces the burden on workers while maintaining high quality work.

Acknowledgements

We would like to thank Judy Kay, Ellen Stuart, Greg Durrett, attendees at the Conference on Human Computation and Crowdsourcing, and the ACL reviewers for helpful feedback. This aterial is based in part on work supported by DARPA (grant #D19AP00079), Bloomberg (Data Science Research Grant), and the Allen Institute for AI (Key Scientific Challenges Program).

References

[alisonlovepowell]. 2015. How long did it take you to hit 5000 completed hits? https://www.reddit.com/r/mturk/comments/3by1va/how_long_did_it_take_you_to_hit_5000_completed/. Accessed: 2020-08-12.

Amazon Mechanical Turk. 2012. Improving quality with qualifications – tips for api requesters. https://blog.mturk.com/improving-quality-with-qualifications-tips-for-api-requesters-87eff638f1d1. Accessed: 2020-08-12.

Amazon Mechanical Turk. 2013. Hit critique: Design tips for improving results. https://blog.mturk.com/hit-critique-design-tips-for-improving-results-a53eb8422081. Accessed: 2020-08-12.

Amazon Mechanical Turk. 2017. Tutorial: Understanding requirements and qualifications. https://blog.mturk.com/tutorial-understanding-requirements-and-qualifications-99a26069fba2. Accessed: 2020-08-12.

Amazon Mechanical Turk. 2019. Qualifications and worker task quality. https://blog.mturk.com/qualifications-and-worker-task-quality-best-practices-886f1f4e03fc. Accessed: 2020-08-12.

Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S. Lasecki, Daniel S. Weld, and Eric Horvitz. 2019. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the Seventh AAAI Conference on Human Computation and Crowdsourcing*.

Benjamin B. Bederson and Alexander J. Quinn. 2011. Web workers unite! addressing challenges of online laborers. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, pages 97–106.

[CaptainSlop]. 2019. Newbie that read faq's any tips to getting to 1000. https://www.reddit.com/r/mturk/comments/9bfv92/newbie_that_read_faqs_any_tips_to_getting_to_1000/e52rbs5/. Accessed: 2020-08-12.

[clickhappier]. 2016. Masters qualification info - everything you need to know. https://www.mturkcrowd.com/threads/masters-qualification-info-everything-you-need-to-know.1453/. Accessed: 2020-08-12.

[Crazybritzombie]. 2018. How to get to 5,000 approved hits? https://www.reddit.com/r/mturk/comments/90zzt5/how_to_get_to_5000_approved_hits/. Accessed: 2020-08-12.

Nerisa Dozo. 2020. Introduction to mturk and prolific.

Kinda El Maarry, Kristy Milland, and Wolf-Tilo Balke. 2018. A fair share of the work? the evolving ecosystem of crowd workers. In *Proceedings of the 10th ACM Conference on Web Science*, page 145–152.

Taylor Nicole Carlson (née Feenstra). 2014. Mechanical turk how to guide. http://pages.ucsd.edu/~tfeenstr/resources/mturkhowto.pdf. Accessed: 2020-08-12.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Last words: Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420.

[FrobozzYogurt]. 2020. Just hit 100k! https://www.reddit.com/r/mturk/comments/i3nvx4/just_hit_100k/. Accessed: 2020-08-12.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

[GnomeWaiter]. 2013. Over $1100 and 5000+ approvals in my first month of turking, and so can you! https://www.reddit.com/r/mturk/comments/1tjge3/over_1100_and_5000_approvals_in_my_first_month_of/. Accessed: 2020-08-12.

Lei Han, Kevin Roitero, Ujwal Gadiraju, Cristina Sarasua, Alessandro Checco, Eddy Maddalena, and Gianluca Demartini. 2019. All those wasted hours: On task abandonment in crowdsourcing. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 321–329.

Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P. Bigham. 2018. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 1–14.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

Lilly C. Irani and M. Six Silberman. 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 611–620.

Lilly C. Irani and M. Six Silberman. 2016. Stories we tell about labor: Turkopticon and the trouble with "design". In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4573–4586.

[jklmnop]. 2016. Your first 1000 hits. https://www.mturkcrowd.com/threads/your-first-1000-hits.23/. Accessed: 2020-08-12.

Neeraj Kumar. 2014. Effective use of amazon mechanical turk (mturk). https://neerajkumar.org/writings/mturk/. Accessed: 2020-08-12.

Pietro Michelucci Libuše Hannah Vepřek, Patricia Seymour. 2020. Human computation requires and enables a new approach to ethical review. In *Proceedings of the NeurIPS Crowd Science Workshop*.

David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a turker. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 224–235.

Joe Miele. 2012. Tips for academic requesters on mturk. http://turkrequesters.blogspot.com/2012/09/tips-for-academic-requesters-on-mturk.html. Accessed: 2020-08-12.

Joe Miele. 2018. The bot problem on mturk. http://turkrequesters.blogspot.com/2018/08/the-bot-problem-on-mturk.html. Accessed: 2020-08-12.

Kristy Milland. 2016. The best practices of the best requesters. http://crowdsourcing-class.org/slides/best-practices-of-best-requesters.pdf. Accessed: 2020-08-12.

Tanushree Mitra, C.J. Hutto, and Eric Gilbert. 2015. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, page 1345–1354.

Jonas Oppenlaender, Kristy Milland, Aku Visuri, Panos Ipeirotis, and Simo Hosio. 2020. Creativity on paid crowdsourcing platforms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–14.

Jonathan Robinson. 2015. Maximizing hit participation. https://www.cloudresearch.com/resources/blog/maximizing-hit-participation/. Accessed: 2020-08-12.

Paul Roit, Ayal Klein, Daniela Stepanov, Jonathan Mamou, Julian Michael, Gabriel Stanovsky, Luke Zettlemoyer, and Ido Dagan. 2020. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866.

Boaz Shmueli, Jan Fell, Soumya Ray, and Lun-Wei Ku. 2021. Beyond fair pay: Ethical implications of NLP crowdsourcing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3758–3769.

Yan Shvartzshnaid, Noah Apthorpe, Nick Feamster, and Helen Nissenbaum. 2019. Going against the (appropriate) flow: A contextual integrity approach to privacy policy analysis. In *Proceedings of the Seventh AAAI Conference on Human Computation and Crowdsourcing*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Colin Vandenhof. 2019. A hybrid approach to identifying unknown unknowns of predictive models. In *Proceedings of the Seventh AAAI Conference on Human Computation and Crowdsourcing*.

[Wat3rloo]. 2016. 5000 approved hits!!?? https://www.reddit.com/r/mturk/comments/4kd1co/5000_approved_hits/. Accessed: 2020-08-12.

[WhereIsTheWork]. 2019. How important are qualifications for getting more surveys? https://www.mturkcrowd.com/threads/how-important-are-qualifications-for-getting-more-surveys.4521/. Accessed: 2020-08-12.

Mark E. Whiting, Grant Hugh, and Michael S. Bernstein. 2019. Fair work: Crowd work minimum wage with one line of code. In *Proceedings of the Seventh AAAI Conference on Human Computation and Crowdsourcing*.

Jacob Young and Kristie M. Young. 2019. Don't get lost in the crowd: Best practices for using amazon's mechanical turk in behavioral research. *Journal of the Midwest Association for Information Systems (JMWAIS)*.

Men Are Elected, Women Are Married: Events Gender Bias on Wikipedia

Jiao Sun[1,2] and **Nanyun Peng**[1,2,3]
[1]Computer Science Department, University of Southern California
[2]Information Sciences Institute, University of Southern California
[3]Computer Science Department, University of California, Los Angeles
jiaosun@usc.edu, violetpeng@ucla.cs.edu

Abstract

Human activities can be seen as sequences of events, which are crucial to understanding societies. Disproportional event distribution for different demographic groups can manifest and amplify social stereotypes, and potentially jeopardize the ability of members in some groups to pursue certain goals. In this paper, we present the first event-centric study of gender biases in a Wikipedia corpus. To facilitate the study, we curate a corpus of career and personal life descriptions with demographic information consisting of 7,854 fragments from 10,412 celebrities. Then we detect events with a state-of-the-art event detection model, calibrate the results using strategically generated templates, and extract events that have asymmetric associations with genders. Our study discovers that Wikipedia pages tend to intermingle personal life events with professional events for females but not for males, which calls for the awareness of the Wikipedia community to formalize guidelines and train the editors to mind the implicit biases that contributors carry. Our work also lays the foundation for future works on quantifying and discovering event biases at the corpus level.

1 Introduction

Researchers have been using NLP tools to analyze corpora for various tasks on online platforms. For example, Pei and Jurgens (2020) found that female-female interactions are more intimate than male-male interactions on Twitter and Reddit. Different from social media, open collaboration communities such as Wikipedia have slowly won the trust of public (Young et al., 2016). Wikipedia has been trusted by many, including professionals in work tasks such as scientific journals (Kousha and Thelwall, 2017) and public officials in powerful positions of authority such as court briefs (Gerken, 2010). Implicit biases in such knowledge sources could have a significant impact on audiences' perception of different groups, thus propagating and even amplifying societal biases. Therefore, analyzing potential biases in Wikipedia is imperative.

Name	Wikipedia Description
Loretta Young (F)	**Career**: In 1930, when she was 17, she eloped with 26-year-old actor Grant Withers; they were married in Yuma, Arizona. The marriage was annulled the next year, just as their second movie together (ironically entitled Too Young to Marry) was released .
Grant Withers (M)	**Personal Life**: In 1930, at 26, he eloped to Yuma, Arizona with 17-year-old actress Loretta Young. The marriage ended in annulment in 1931 just as their second movie together, titled Too Young to Marry, was released .

Table 1: The marriage events are under the *Career* section for the female on Wikipedia. However, the same marriage is in the *Personal Life* section for the male. yellow background highlights events in the passage.

In particular, studying events in Wikipedia is important. An event is a specific occurrence under a certain time and location that involves participants (Yu et al., 2015); human activities are essentially sequences of events. Therefore, the distribution and perception of events shape the understanding of society. Rashkin et al. (2018) discovered implicit gender biases in film scripts using events as a lens. For example, they found that events with female agents are intended to be helpful to other people, while events with male agents are motivated by achievements. However, they focused on the intentions and reactions of events rather than events themselves.

In this work, we propose to use events as a lens to study gender biases and demonstrate that events are more efficient for understanding biases in corpora than raw texts. We define *gender bias* as the

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 350–360
August 1–6, 2021. ©2021 Association for Computational Linguistics

asymmetric association of events with females and males,[1] which may lead to gender stereotypes. For example, females are more associated with domestic activities than males in many cultures (Leopold, 2018; Jolly et al., 2014).

To facilitate the study, we collect a corpus that contains demographic information, personal life description, and career description from Wikipedia.[2] We first detect events in the collected corpus using a state-of-the-art event extraction model (Han et al., 2019). Then, we extract gender-distinct events with a higher chance to occur for one group than the other. Next, we propose a calibration technique to offset the potential confounding of gender biases in the event extraction model, enabling us to focus on the gender biases at the corpus level. Our contributions are three-fold:

- We contribute a corpus of 7,854 fragments from 10,412 celebrities across 8 occupations including their demographic information and Wikipedia *Career* and *Personal Life* sections.
- We propose using events as a lens to study gender biases at the corpus level, discover a mixture of personal life and professional life for females but not for males, and demonstrate the efficiency of using events in comparison to directly analyzing the raw texts.
- We propose a generic framework to analyze event gender bias, including a calibration technique to offset the potential confounding of gender biases in the event extraction model.

2 Experimental Setup

In this section, we will introduce our collected corpus and the event extraction model in our study.

Dataset. Our collected corpus contains demographics information and description sections of celebrities from Wikipedia. Table 2 shows the statistics of the number of celebrities with *Career* or *Personal Life* sections in our corpora, together with all celebrities we collected. In this work, we only explored celebrities with *Career* or *Personal Life* sections, but there are more sections (e.g., *Politics* and *Background and Family*) in our collected corpus. We encourage interested researchers to further utilize our collected corpus and conduct studies from other perspectives. In each experiment, we select the same number of female and male celebrities from one occupation for a fair comparison.

	Career		Personal Life		Collected	
Occ	F	M	F	M	F	M
Acting	464	469	464	469	464	469
Writer	455	611	319	347	1,372	2,466
Comedian	380	655	298	510	642	1,200
Artist	193	30	60	18	701	100
Chef	81	141	72	95	176	350
Dancer	334	167	286	127	812	465
Podcaster	87	183	83	182	149	361
Musician	39	136	21	78	136	549
All	4,425		3,429		10,412	

Table 2: Statistics showing the number of celebrities with *Career* section or *Personal Life* section, together with all celebrities we collected. Not all celebrities have *Career* or *Personal Life* sections.

Event Extraction. There are two definitions of events: one defines an event as the trigger word (usually a verb) (Pustejovsky et al., 2003b), the other defines an event as a complex structure including a trigger, arguments, time, and location (Ahn, 2006). The corpus following the former definition usually has much broader coverage, while the latter can provide richer information. For broader coverage, we choose a state-of-the-art event detection model that focuses on detecting event trigger words by Han et al. (2019).[3] We use the model trained on the TB-Dense dataset (Pustejovsky et al., 2003a) for two reasons: 1) the model performs better on the TB-Dense dataset; 2) the annotation of the TB-Dense dataset is from the news articles, and it is also where the most content of Wikipedia comes from.[4] We extract and lemmatize events e from the corpora and count their frequencies $|e|$. Then, we separately construct dictionaries $\mathcal{E}^m = \{e_1^m : |e_1^m|, ..., e_M^m : |e_M^m|\}$ and $\mathcal{E}^f = \{e_1^f : |e_1^f|, ..., e_F^f : |e_F^f|\}$ mapping events to their frequency for male and female respectively.

Event Extraction Quality. To check the model performance on our corpora, we manually annotated events in 10,508 sentences (female: 5,543,

[1] In our analysis, we limit to binary gender classes, which, while unrepresentative of the real-world diversity, allows us to focus on more depth in analysis.

[2] https://github.com/PlusLabNLP/ee-wiki-bias

[3] We use the code at https://github.com/rujunhan/EMNLP-2019 and reproduce the model trained on the TB-Dense dataset.

[4] According to Fetahu et al. (2015), more than 20% of the references are news articles on Wikipedia.

Metric	TB-D	S	S-F	S-M
Precision	89.2	93.5	95.3	93.4
Recall	92.6	89.8	87.1	89.8
F1	90.9	91.6	91.0	91.6

Table 3: The performance for off-the-shelf event extraction model in both common event extraction dataset TB-Dense (TB-D) and our corpus with manual annotation. `S` represents the sampled data from the corpus. `S-F` and `S-M` represent the sampled data for female career description and male career description separately.

male: 4,965) from the Wikipedia corpus. Table 3 shows that the model performs comparably on our corpora as on the TB-Dense test set.

3 Detecting Gender Biases in Events

Odds Ratio. After applying the event detection model, we get two dictionaries $\mathcal{E}^m$ and $\mathcal{E}^f$ that have events as keys and their corresponding occurrence frequencies as values. Among all events, we focus on those with distinct occurrences in males and females descriptions (e.g., `work` often occurs at a similar frequency for both females and males in *Career* sections, and we thus neglect it from our analysis). We use the Odds Ratio (OR) (Szumilas, 2010) to find the events with large frequency differences for females and males, which indicates that they might potentially manifest gender biases. For an event e_n, we calculate its odds ratio as the odds of having it in the male event list divided by the odds of having it in the female event list:

$$\frac{\mathcal{E}^m(e_n)}{\sum^{i}_{\substack{e_i^m \neq e_n \\ i\in[1,\ldots,M]}} \mathcal{E}^m(e_i^m)} \Big/ \frac{\mathcal{E}^f(e_n)}{\sum^{j}_{\substack{e_j^f \neq e_n \\ j\in[1,\ldots,F]}} \mathcal{E}^f(e_j^f)} \quad (1)$$

The larger the OR is, the more likely an event will occur in male than female sections by Equation 1. After obtaining a list of events and their corresponding OR, we sort the events by OR in descending order. The top k events are more likely to appear for males and the last k events for females.

Calibration. The difference of event frequencies might come from the model bias, as shown in other tasks (e.g., gender bias in coreference resolution model (Zhao et al., 2018)). To offset the potential confounding that could be brought by the event extraction model and estimate the actual event frequency, we propose a calibration strategy by 1) generating data that contains target events; 2) testing the model performance for females and males separately in the generated data, 3) and using the model performance to estimate real event occurrence frequencies.

We aim to calibrate the top 50 most skewed events in females' and males' *Career* and *Personal Life* descriptions after using the OR separately. First, we follow two steps to generate a synthetic dataset:

1. For each target event, we select all sentences where the model successfully detected the target event. For each sentence, we manually verify the correctness of the extracted event and discard the incorrect ones. For the rest, we use the verified sentences to create more ground truth; we call them *template sentences*.

2. For each template sentence, we find the celebrity's first name and mark it as a `Name Placeholder`, then we replace it with 50 female names and 50 male names that are sampled from the name list by Ribeiro et al. (2020). If the gender changes during the name replacement (e.g., Mike to Emily), we replace the corresponding pronouns (e.g., he to she) and gender attributes (Zhao et al., 2018) (e.g., Mr to Miss) in the template sentences. As a result, we get 100 data points for each template sentence with automatic annotations. If there is no first name in the sentence, we replace the pronouns and gender attributes.

After getting the synthetic data, we run the event extraction model again. We use the detection recall among the generated instances to calibrate the frequency $|e|$ for each target event and estimate the actual frequency $|e|^*$, following:

$$|e|^* = \frac{|e|}{TP(e)/(TP(e)+FP(e))} \quad (2)$$

Then, we replace $|e|$ with $|e|^*$ in Equation 1, and get k female and k male events by sorting OR as before. Note that we observe the model performances are mostly unbiased, and we have only calibrated events that have different performances for females and males over a threshold (i.e., 0.05).[6]

[5]ACE dataset: https://www.ldc.upenn.edu/collaborations/past-projects/ace

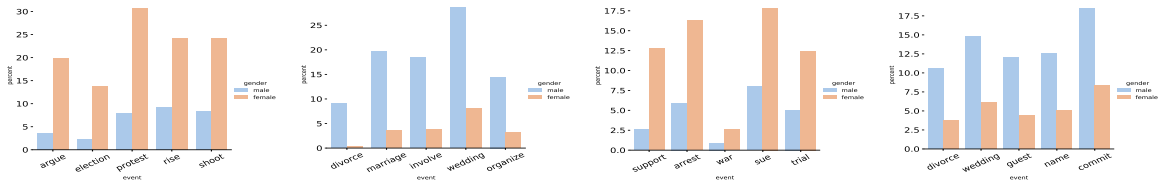
[5]We did not show the result for the `artists` and `musicians` due to the small data size.

[6]Calibration details and quantitative result in App. A.2.

Occupation	Events in Female Career Description	Events in Male Career Description	WEAT∗	WEAT
Writer	♦ divorce, ♦ marriage, involve, organize, ♦ wedding	argue, ⊕ election, ▲ protest, rise, shoot	-0.17	1.51
Acting	♦ divorce, ♦ wedding, guest, name, commit	support, ▲ arrest, ▲ war, ■ sue, trial	-0.19	0.88
Comedian	♦ birth, eliminate, ♦ wedding, ♥ relocate, partner	enjoy, hear, cause, ● buy, conceive	-0.19	0.54
Podcaster	♥ land, interview, portray, ♦ married, report	direct, ask, provide, continue, bring	-0.24	0.53
Dancer	♦ married, ♦ marriage, ♥ depart, ♥ arrive, organize	drop, team, choreograph, explore break	-0.14	0.22
Artist	paint, exhibit, include, ♥ return, teach	start, found, feature, award, begin	-0.02	0.17
Chef	⊕ hire, △ meet, debut, eliminate, sign	include, focus, explore, award, ● raise	-0.13	-0.38
Musician	run, record, ♦ death, found, contribute	sign, direct, produce, premier, open	-0.19	–0.41

Annotations: ♦ Life ♥ Transportation ⊕ Personell ▲ Conflict ■ Justice ● Transaction △ Contact

Table 4: Top 5 extracted events that occur more often for females and males in *Career* sections across 8 occupations. We predict event types by applying EventPlus (Ma et al., 2021) on sentences that contain target events and take the majority vote of the predicted types. The event types are from the ACE dataset.[5] We calculate WEAT scores with all tokens excluding stop words (WEAT* column) and only detected events (WEAT column) for *Career* sections.



(a) Male Writers (b) Female Writers (c) Actor (d) Actress

Figure 1: The percentile of extracted events among all detected events, sorted by their frequencies in descending order. The smaller the percentile is, the more frequent the event appears in the text. The extracted events are among the top 10% for the corresponding gender (e.g., extracted female events among all detected events for female writers) and within top 40% percent for the opposite gender (e.g., extracted female events among all detected events for male writers). The figure shows that we are not picking rarely-occurred events, and the result is significant.

WEAT score. We further check if the extracted events are associated with gender attributes (e.g., `she and her` for females, and `he and him` for males) in popular neural word embeddings like Glove (Pennington et al., 2014). We quantify this with the Word Embedding Association Test (WEAT) (Caliskan et al., 2017), a popular method for measuring biases in text. Intuitively, WEAT takes a list of tokens that represent a concept (in our case, *extracted events*) and verifies whether these tokens have a shorter distance towards female attributes or male attributes. A positive value of WEAT score indicates that female events are closer to female attributes, and male events are closer to male attributes in the word embedding, while a negative value indicates that female events are closer to male attributes and vice versa.[7]

To show the effectiveness of using events as a lens for gender bias analysis, we compute WEAT scores on the raw texts and detected events separately. For the former, we take all tokens excluding stop words.[8] Together with gender attributes from Caliskan et al. (2017), we calculate and show the WEAT scores under two settings as "WEAT*" for the raw texts and "WEAT" for the detected events.

4 Results

The Effectiveness of our Analysis Framework. Table 4 and Table 5 show the associations of both raw texts and the extracted events in *Career* and *Personal Life* sections for females and males across occupations after the calibration. The values in WEAT* columns in both tables indicate that there

[7]Details of WEAT score experiment in App. A.4.

[8]We use spaCy (`https://spacy.io/`) to tokenize the corpus and remove stop words.

Occupation	Events in Female Personal Life Description	Events in Male Personal Life Description	WEAT∗	WEAT
Writer	bury, ♦ birth, attend, ▲ war, grow	know, report, come, ■ charge, publish	-0.05	0.31
Acting	♦ pregnant, practice, wedding, record, convert	accuse, ♥ trip, ♥ fly, ▲ assault, endorse	-0.14	0.54
Comedian	feel, ♦ birth, fall, open, decide	♥ visit, create, spend, propose, lawsuit	-0.07	0.07
Podcaster	date, describe, tell, life, come	play, write, ♦ born, release, claim	-0.13	0.57
Dancer	♦ marry, describe, diagnose, expect, speak	hold, involve, ● award, run, serve	-0.03	0.41
Chef	♦ death, serve, announce, describe, ♦ born	♦ birth, lose, ♦ divorce, speak, △ meet	-0.02	-0.80
Annotations: ♦ Life ♥ Transportation ⊕ Personell ▲ Conflict ■ Justice ● Transaction △ Contact				

Table 5: Top 5 events in *Personal Life* section across 6 occupations.[9] There are more `Life` events (e.g., "birth" and "marry") in females' personal life descriptions than males' for most occupations. While for males, although we see more life-related events than in the *Career* section, there are events like "awards" even in the *Personal Life* section. The findings further show our work is imperative and addresses the importance of not intermingling the professional career with personal life regardless of gender during the future editing on Wikipedia.

was only a weak association of words in raw texts with gender. In contrast, the extracted events are associated with gender for most occupations. It shows the effectiveness of the event extraction model and our analysis method.

The Significance of the Analysis Result. There is a possibility that our analysis, although it picks out distinct events for different genders, identifies the events that are infrequent for all genders and that the frequent events have similar distributions across genders. To verify, we sort all detected events from our corpus by frequencies in descending order. Then, we calculate the percentile of extracted events in the sorted list. The smaller the percentile is, the more frequent the event appears in the text. Figure 1 shows that we are not picking the events that rarely occur, which shows the significance of our result.[10] For example, Figure 1a and Figure 1b show the percentile of frequencies for selected male and female events among all events frequencies in the descending order for male and female writers, respectively. We can see that for the corresponding gender, event frequencies are among the top 10%. These events occur less frequently for the opposite gender but still among the top 40%.

Findings and Discussions. We find that there are more `Life` events for females than males in both *Career* and *Personal Life* sections. On the other hand, for males, there are events like "awards" even in their *Personal Life* section. The mixture of personal life with females' professional career events and career achievements with males' personal life events carries implicit gender bias and reinforces the gender stereotype. It potentially leads to career, marital, and parental status discrimination towards genders and jeopardizes gender equality in society. We recommend: 1) Wikipedia editors to restructure pages to ensure that personal life-related events (e.g., marriage and divorce) are written in the *Personal Life* section, and professional events (e.g., award) are written in *Career* sections regardless of gender; 2) future contributors should also be cautious and not intermingle *Personal Life* and *Career* when creating the Wikipedia pages from the start.

[10] See plots for all occupations in Appendix A.5.

5 Conclusion

We conduct the first event-centric gender bias analysis at the corpus level and compose a corpus by scraping Wikipedia to facilitate the study. Our analysis discovers that the collected corpus has event gender biases. For example, personal life related events (e.g., `marriage`) are more likely to appear for females than males even in *Career* sections. We hope our work brings awareness of potential gender biases in knowledge sources such as Wikipedia, and urges Wikipedia editors and contributors to be cautious when contributing to the pages.

Acknowledgments

This material is based on research supported by IARPA BETTER program via Contract No. 2019-19051600007. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government.

Ethical Considerations

Our corpus is collected from Wikipedia. The content of personal life description, career description, and demographic information is all public to the general audience. Note that our collected corpus might be used for malicious purposes. For example, it can serve as a source by text generation tools to generate text highlighting gender stereotypes.

This work is subject to several limitations: First, it is important to understand and analyze the event gender bias for gender minorities, missing from our work because of scarce resources online. Future research can build upon our work, go beyond the binary gender and incorporate more analysis. Second, our study focuses on the Wikipedia pages for celebrities for two additional reasons besides the broad impact of Wikipedia: 1) celebrities' Wikipedia pages are more accessible than non-celebrities. Our collected Wikipedia pages span across 8 occupations to increase the representation of our study; 2) Wikipedia contributors have been extensively updating celebrities' Wikipedia pages every day. Wikipedia develops at a rate of over 1.9 edits every second, performed by editors from all over the world (wik, 2021). The celebrities' pages get more attention and edits, thus better present how the general audience perceives important information and largely reduce the potential biases that could be introduced in personal writings. Please note that although we try to make our study as representative as possible, it cannot represent certain groups or individuals' perceptions.

Our model is trained on TB-Dense, a public dataset coming from news articles. These do not contain any explicit detail that leaks information about a user's name, health, negative financial status, racial or ethnic origin, religious or philosophical affiliation or beliefs, trade union membership, alleged or actual crime commission.

References

2021. Wikipedia:statistics - wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Statistics. (Accessed on 02/01/2021).

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

A. Caliskan, J. Bryson, and A. Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356:183 – 186.

Besnik Fetahu, Katja Markert, and Avishek Anand. 2015. Automated news suggestions for populating wikipedia entity pages. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 323–332.

Joseph L. Gerken. 2010. How courts use wikipedia. *The Journal of Appellate Practice and Process*, 11:191.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China. Association for Computational Linguistics.

S. Jolly, K. Griffith, R. Decastro, A. Stewart, P. Ubel, and R. Jagsi. 2014. Gender differences in time spent on parenting and domestic responsibilities by high-achieving young physician-researchers. *Annals of Internal Medicine*, 160:344–353.

K. Kousha and M. Thelwall. 2017. Are wikipedia citations important evidence of the impact of scholarly articles and books? *Journal of the Association for Information Science and Technology*, 68.

T. Leopold. 2018. Gender differences in the consequences of divorce: A study of multiple outcomes. *Demography*, 55:769 – 797.

Mingyu Derek Ma, Jiao Sun, Mu Yang, Kung-Hsiang Huang, Nuan Wen, Shikhar Singh, Rujun Han, and Nanyun Peng. 2021. Eventplus: A temporal event understanding pipeline. In *2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Demonstrations Track*.

Jiaxin Pei and David Jurgens. 2020. Quantifying intimacy in language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5307–5326, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa

Ferro, et al. 2003b. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. Event2Mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, Melbourne, Australia. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

M. Szumilas. 2010. Explaining odds ratios. *Journal of the Canadian Academy of Child and Adolescent Psychiatry = Journal de l'Academie canadienne de psychiatrie de l'enfant et de l'adolescent*, 19 3:227–9.

A. Young, Ari D. Wigdor, and Gerald Kane. 2016. It's not what you think: Gender bias in information about fortune 1000 ceos on wikipedia. In *ICIS*.

Mo Yu, Matthew R. Gormley, and Mark Dredze. 2015. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1374–1379, Denver, Colorado. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

A.1 Quality Check: Event Detection Model

To test the performance of the event extraction model in our collected corpus from Wikipedia. We manually annotated events in 10,508 (female: 5,543, male: 4,965) sampled sentences from the *Career* section in our corpus. Our annotators are two volunteers who are not in the current project but have experience with event detection tasks. We asked annotators to annotate all event trigger words in the text. During annotation, we follow the definition of events from the ACE annotation guideline.[11] We use the manual annotation as the ground truth and compare it with the event detection model output to calculate the metrics (i.e., precision, recall and F1) in Table 3.

A.2 Calibration Details

To offset the potential confounding that could be brought by the event extraction model and estimate the actual event frequency of $|e|^*$, we use the recall for the event e to calibrate the event frequency $|e|$ for females and males separately. Figure 2 shows the calibration result for the 20 most frequent events in our corpus. Please note that Figure 2 (a)-(h) show the quantitative result for extracted events in the *Career* sections across 8 occupations, and Figure 2 (i)-(n) for the *Personal Life* sections.

Example Sentence Substitutions for Calibration. After checking the quality of selected sentences containing the target event trigger, we use 2 steps described in Section 3 *Calibration* to compose a synthetic dataset with word substitutions. Here is an example of using `Name Placeholder`: for target event trigger "married" in `Carole Baskin`'s *Career* section, we have:

> At the age of 17, Baskin worked at a Tampa department store. To make money, she began breeding show cats; she also began rescuing bobcats, and used llamas for a lawn trimming business. In January 1991, she married her second husband and joined his real estate business.

First, we mark the first name Baskin as `Name Placeholder` and find all gender attributes and pronouns which are consistent with the celebrity's gender. Then, we replace Baskin with 50 female names and 50 male names from Ribeiro et al. (2020). If the new name is a male name, we change the corresponding gender attributes (none in this case) and pronouns (e.g., she to he, her to his).

Another example is for the context containing the target event trigger "married" in `Indrani Rahman`'s *Career* section, where there is no first name:

> In 1952, although married, and with a child, she became the first Miss India, and went on to compete in the Miss Universe 1952 Pageant, held at Long Beach, California. Soon, she was travelling along with her mother and performing all over the world...

We replace all pronouns (she to he, her to his) and gender attributes (Miss to Mr).

Interpret the Quantitative Calibration Result. We use the calibration technique to calibrate potential gender biases from the model that could have complicated the analysis. In Figure 2, we can see that there is little gender bias at the model level: the model has the same performance for females and males among most events.

Besides, we notice that the model fails to detect and has a low recall for few events in the generated synthetic dataset. We speculate that this is because of the brittleness in event extraction models triggered by the word substitution. We will leave more fine-grained analysis at the model level for future work. We focus on events for which the model performs largely different for females and males during our calibration. Thus, we select and focus on the events that have different performance for females and males over a threshold, which we take 0.05 during our experiment, to calibrate the analysis result.

A.3 Top Ten Extracted Events

Table 6 and Table 7 show the top 10 events and serves as the supplement of top 5 events that we reported for *Career* and *Personal Life* sections.

A.4 Details for Calculating WEAT Score

The WEAT score is in the range of -2 to 2. A high positive score indicates that extracted events for females are more associated with female attributes in the embedding space. A high negative score means that extracted events for females are more

[11] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

Occupation	Events in Female Career Description	Events in Male Career Description
Writer	divorce, marriage, involve, organize, wedding, donate, fill, pass, participate, document	argue, election, protest, rise, shoot, purchase, kill, host, close, land
Acting	divorce, wedding, guest, name, commit, attract, suggest, married, impressed, induct	support, arrest, war, sue, trial, vote, pull, team, insist, like
Comedian	birth, eliminate, wedding, relocate, partner, pursue, impersonate, audition, guest, achieve	enjoy, hear, cause, buy, conceive, enter, injury, allow, acquire, enter
Podcaster	land, interview, portray, married, report, earn, praise, talk, shoot, premier	direct, ask, provide, continue, bring, election, sell, meet, read, open
Dancer	married, marriage, depart, arrive, organize try, promote, train, divorce, state	drop, team, choreograph, explore, break, think, add, celebrate, injury, suffer
Artist	paint, exhibit, include, return, teach, publish, explore, draw, produce, write	start, found, feature, award, begin, appear, join, influence, work, create
Chef	hire, meet, debut, eliminate, sign, graduate, describe, train, begin, appear	include, focus, explore, award, raise, gain, spend, find, launch, hold
Musician	run, record, death, found, contribute, continue, perform, teach, appear, accord	sign, direct, produce, premier, open, announce, follow, star, act, write

Table 6: The top 10 extracted events in *Career* section.

Occupation	Events in Female Personal Life Description	Events in Male Personal Life Description
Writer	bury, birth, attend, war, grow, serve, appear, raise, begin, divorce	know, report, come, charge, publish, claim, suffer, return, state, describe
Acting	pregnant, practice, wedding, record, convert, honor, gain, retire, rap, bring	accuse, trip, fly, assault, endorse, meeting, donate, fight, arrest, found
Comedian	feel, birth, fall, open, decide, date, diagnose, tweet, study, turn	visit, create, spend, propose, lawsuit, accord, arrest, find, sell, admit
Podcaster	date, describe, tell, life, come, leave, engage, live, start, reside	play, write, bear, release, claim, birth, divorce, meet, announce, work
Dancer	marry, describe, diagnose, expect, speak, post, attend, come, play, reside	hold, involve, award, run, serve, adopt, charge, suit, struggle, perform
Chef	death, serve, announce, describe, born, die, life, state, marriage, live	birth, lose, divorce, speak, meet, work, diagnose, wedding, write, engage

Table 7: The top 10 extracted events in *Personal Life* section.

associated with male attributes. To calculate the WEAT score, we input two lists of extracted events for females E_f and males E_m, together with two lists of gender attributes A and B, then calculate:

$$S(E_f, E_m, A, B) = \sum_{e_f \in E_f} s(e_f, A, B) - \sum_{e_m \in E_m} s(e_m, A, B), \tag{3}$$

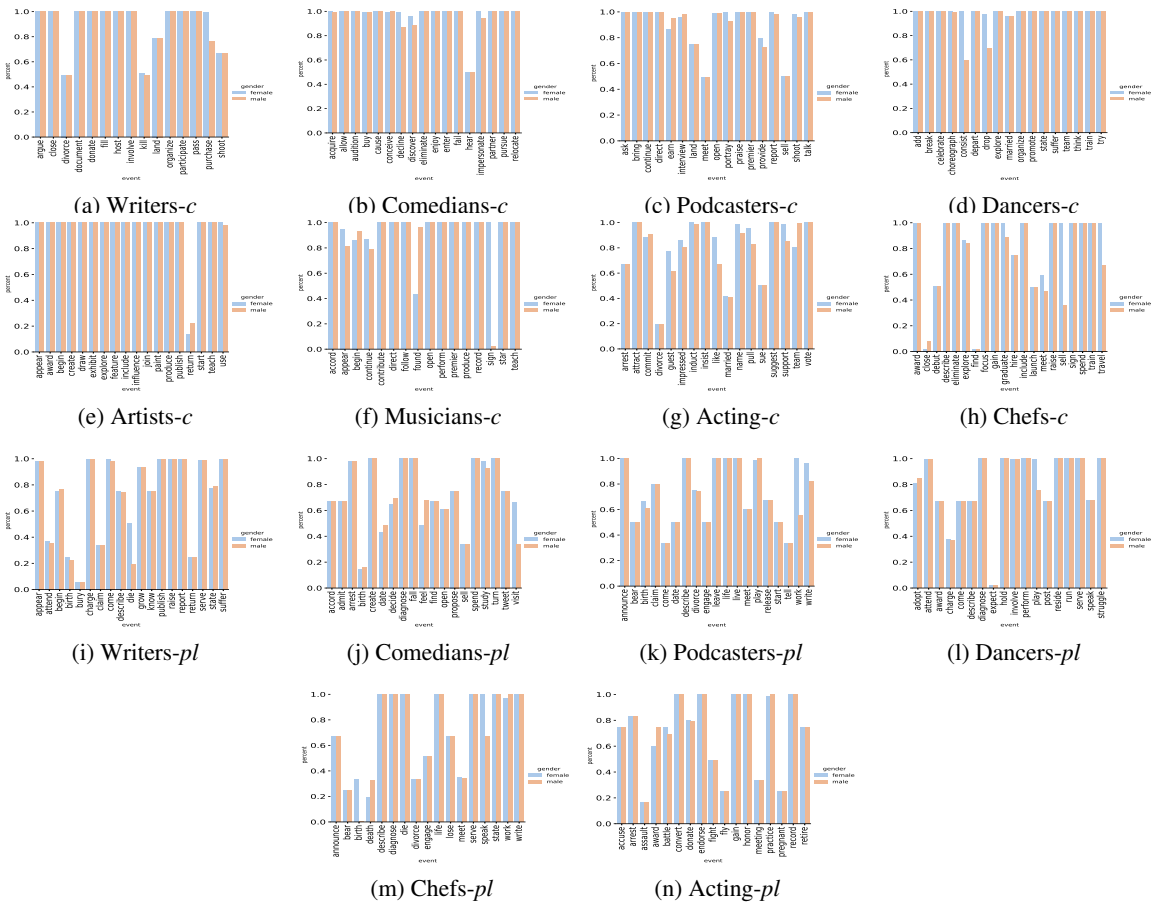
where

$$s(w, A, B) = \text{mean}_{a \in A} cos(\vec{w}, \vec{a}) - \text{mean}_{b \in B} cos(\vec{w}, \vec{b}). \tag{4}$$

Following Caliskan et al. (2017), we have "female, woman, girl, sister, she, her, hers, daughter" as female attribute list A and "male, man, boy, brother, he, him, his, son" as male attributes list B. To calculate WEAT*, we replace the input lists E_f and E_m with all non-stop words tokens in raw texts from either *Career* section or *Personal Life* section.

A.5 Extracted Events Frequency Distribution

We sort all detected events from our corpus by their frequencies in descending order according to Equation 1. Figure 3 (a)-(l) show the percentile of extracted events in the sorted list for another 6 occupations besides the 2 occupations reported in Figure 1 for *Career* section. The smaller the percentile is, the more frequent the event appears in the text. These figures indicate that we are not picking events that rarely occur and showcase the significance of our analysis result. Figure 3 (m)-(x) are for *Personal Life* sections across 6 occupations, which show the same trend as for *Career* sections.



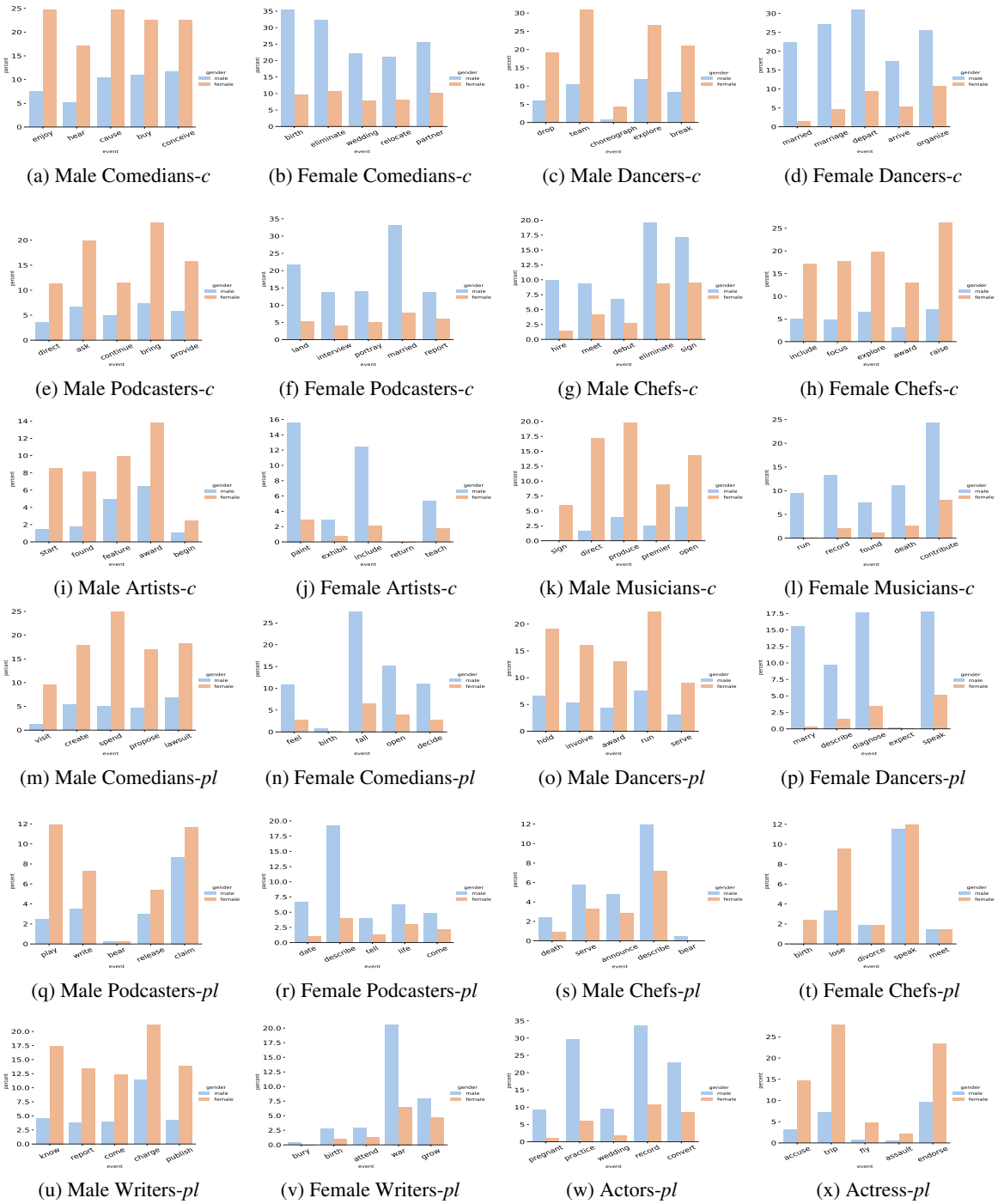
(a) Writers-*c* (b) Comedians-*c* (c) Podcasters-*c* (d) Dancers-*c*

(e) Artists-*c* (f) Musicians-*c* (g) Acting-*c* (h) Chefs-*c*

(i) Writers-*pl* (j) Comedians-*pl* (k) Podcasters-*pl* (l) Dancers-*pl*

(m) Chefs-*pl* (n) Acting-*pl*

Figure 2: Detection recall on the strategically-generated data. (*c*: *Career* section, *pl*: *Personal Life* section)



(a) Male Comedians-*c* (b) Female Comedians-*c* (c) Male Dancers-*c* (d) Female Dancers-*c*

(e) Male Podcasters-*c* (f) Female Podcasters-*c* (g) Male Chefs-*c* (h) Female Chefs-*c*

(i) Male Artists-*c* (j) Female Artists-*c* (k) Male Musicians-*c* (l) Female Musicians-*c*

(m) Male Comedians-*pl* (n) Female Comedians-*pl* (o) Male Dancers-*pl* (p) Female Dancers-*pl*

(q) Male Podcasters-*pl* (r) Female Podcasters-*pl* (s) Male Chefs-*pl* (t) Female Chefs-*pl*

(u) Male Writers-*pl* (v) Female Writers-*pl* (w) Actors-*pl* (x) Actress-*pl*

Figure 3: The percentile of extracted event frequencies. (*c*: *Career* section, *pl*: *Personal Life* section)

Modeling Task-Aware MIMO Cardinality for Efficient Multilingual Neural Machine Translation

Hongfei Xu[1] **Qiuhui Liu**[2] **Josef van Genabith**[1] **Deyi Xiong**[3,4*]
[1]DFKI and Saarland University, Informatics Campus, Saarland, Germany
[2]China Mobile Online Services, Henan, China
[3]Tianjin University, Tianjin, China
[4]Global Tone Communication Technology Co., Ltd.
{hfxunlp, liuqhano}@foxmail.com, josef.van_genabith@dfki.de, dyxiong@tju.edu.cn

Abstract

Neural machine translation has achieved great success in bilingual settings, as well as in multilingual settings. With the increase of the number of languages, multilingual systems tend to underperform their bilingual counterparts. Model capacity has been found crucial for massively multilingual NMT to support language pairs with varying typological characteristics. Previous work increases the modeling capacity by deepening or widening the Transformer. However, modeling cardinality based on aggregating a set of transformations with the same topology has been proven more effective than going deeper or wider when increasing capacity. In this paper, we propose to efficiently increase the capacity for multilingual NMT by increasing the cardinality. Unlike previous work which feeds the same input to several transformations and merges their outputs into one, we present a Multi-Input-Multi-Output (MIMO) architecture that allows each transformation of the block to have its own input. We also present a task-aware attention mechanism to learn to selectively utilize individual transformations from a set of transformations for different translation directions. Our model surpasses previous work and establishes a new state-of-the-art on the large scale OPUS-100 corpus while being 1.31 times as fast.

1 Introduction

Multilingual translation between multiple language pairs with a single model (Firat et al., 2016a; Johnson et al., 2017; Aharoni et al., 2019) has some advantages compared to bilingual systems (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017; Barrault et al., 2020), e.g., easy deployment, enabling transfer learning across languages and zero-shot translation.

Despite their advantages, multilingual systems tend to underperform their bilingual counterparts as the number of languages increases (Johnson et al., 2017; Aharoni et al., 2019). This is due to the fact that multilingual NMT must distribute its modeling capacity over different translation directions. Zhang et al. (2020) show that the model capacity is crucial for massively multilingual NMT to support language pairs with varying typological characteristics, and propose to increase the modeling capacity by deepening the Transformer.

However, compared to going deeper or wider, modeling cardinality based on aggregating a set of transformations with the same topology has been proven more effective when we increase the model capacity (Xie et al., 2017). In this paper, we efficiently increase the capacity of the multilingual NMT model by increasing the cardinality, i.e. stacking sub-layers that aggregate a set of transformations with the same topology.

Our main contributions are as follows:

- We propose to efficiently increase the capacity of the multilingual NMT model by increasing cardinality, and present a novel MIMO design that allows transformations in the subsequent layer to take different outputs from the current layer as their inputs, unlike previous studies (Xie et al., 2017; Yan et al., 2020) which feed the same input to several transformations and merge their outputs into one;
- We propose to learn a task-aware attention mechanism for the MIMO transformation, allowing the model to weigh different transformations of the set differently for specific translation directions;
- In our experiments on the OPUS-100 corpus, our approach outperforms previous work and

* Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 361–367
August 1–6, 2021. ©2021 Association for Computational Linguistics

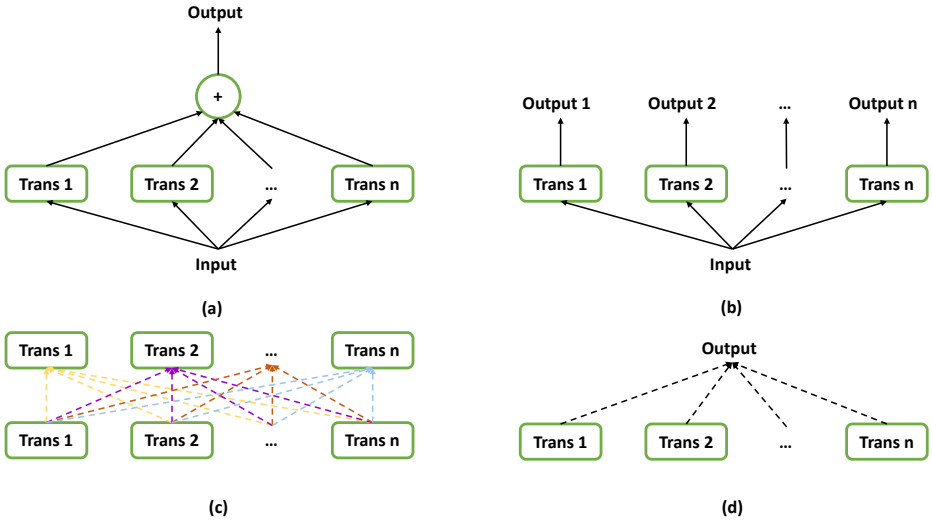


Figure 1: Block transformations. (a) takes the same input into a set of transformations, and adds up their outputs as the output of the block. (b) takes the same input and processes it with these transformations without merging their outputs. (c) is the MIMO architecture that combines weighted outputs of these transformations as inputs to the subsequent transformation set. (d) combines weighted outputs of these transformations into one. Dashed arrows indicate learned attention probabilities. Each "Trans" is a sub-layer that runs in the order of: transforming $\rightarrow$ dropout $\rightarrow$ residual connection $\rightarrow$ layer normalization, where the transforming unit can be either multi-head attention or FFN, as depicted in Figure 2. We aggregate the final output of layer normalization of each "Trans" in the block into the input fed to the next block in different ways (i.e., (a)-(d)).

achieves a new state-of-the-art while being 1.31 times as fast.

2 Preliminaries

Zhang et al. (2020) overcome the capacity bottleneck of multilingual NMT via deepening NMT architectures.

Xie et al. (2017) present a highly modularized network architecture for image classification. The network is constructed by repeating a building block that aggregates a set of transformations with the same topology. For a given input i, the block adopts n networks of the same topology $trans$ to process i and merges their outputs into the final output o of the layer:

$$o = \sum_{k=1}^{n} trans_k(i) \quad (1)$$

This design strategy exposes a new dimension, namely "cardinality" (the size of the set of transformations), as an essential factor in addition to the dimensions of depth and width. Xie et al. (2017) empirically show that increasing cardinality is more effective than going deeper or wider when we increase the capacity to improve classification accuracy.

Yan et al. (2020) present a multi-unit Transformer to efficiently improve the translation performance by increasing cardinality instead of depth. However, their work implements stacks of *input* $\rightarrow$ *performing multiple transformations* $\rightarrow$ *merging* blocks (as illustrated in Figure 1 (a)), is developed for bilingual sentence-level transformation, and requires the additional design of a biasing module and sequential dependency that guide and encourage complementariness among different units. By contrast, our work aims at efficiently increasing the capacity for multilingual translation, proposes the MIMO transformation (Figure 1 (c)) between stacked blocks, and naturally uses the translation task in attention form to guide individual transformations of the set to learn different representations for different translation directions.

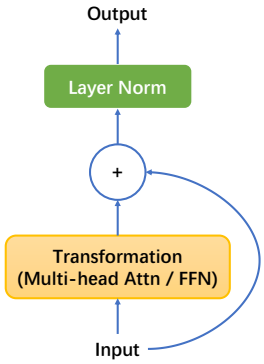


Figure 2: The "Trans" unit.

3 Our Approach

3.1 Multi-Input-Multi-Output (MIMO) Transformation

In contrast to previous approaches (Xie et al., 2017; Yan et al., 2020) that follow a stack of transformation-merging procedures (Figure 1 (a)) to increase cardinality, in our approach we allow our set of transformations to take different inputs. Compared to using the same input, this may encourage transformations to learn complementary representations. Furthermore, merging the outputs of different transformations into one is likely to incur information loss. This is avoided in our approach.

We employ a MIMO transformation between stacked layers (Figure 1 (c)) to enable each transformation of the block to selectively learn to operate on its own unique input.

Specifically, we keep n outputs of the set of transformations to produce multiple inputs for the next layer instead of merging them into one. The input i_k^j to the kth transformation of the jth layer $trans_k^j$ is a weighted accumulation of the outputs o^{j-1} of the layer $j-1$.

$$i_k^j = \sum_{m=1}^{n} p_m^j * o_m^{j-1} \tag{2}$$

where p_m^j are softmax-normalized learnable parameters to model translation task-aware attention for multilingual NMT described in Section 3.2.

o_k^j is produced by $trans_k^j$ with i_k^j as its input:

$$o_k^j = trans_k^j(i_k^j) \tag{3}$$

In the case of a Transformer for multilingual NMT, $trans_k^j$ can be either the multi-head attention or the feed-forward neural network. We adopt a one-to-many transformation (Figure 1 (b)) for the self-attention layer in the first encoder/decoder layer to project one input from the embedding layer to multiple inputs to subsequent layers, and perform a many-to-one transformation (Figure 1 (d)) with the outputs of the feed-forward layer of the last decoder layer to build a single input for the classifier.

3.2 Task-Aware Attention

Rather than separating the multilingual NMT model into 2 parts: 1) the shared part for all language pairs trained on the full dataset; 2) the language isolated part which will only be activated in the corresponding translation task and trained on the part of the whole dataset specifically for the language, we compute all transformations of each block regardless of the translation task, thus all model parameters can utilize and benefit from the whole training set. At the same time, we introduce a task-aware attention mechanism to utilize different transformations of the block differently for specific translation directions.

Specifically, we learn an embedding v for each translation direction (i.e., to X (e.g., en, zh, de)) for each transformation to weightedly aggregate multiple outputs of the block below. v is first normalized into a probability p:

$$p = \mathrm{softmax}(v) \tag{4}$$

Next, p is used in Equation 2 for weighted aggregation. p is expected to assign a higher weight to corresponding transformations of the block which are more important for the translation direction.

3.3 Discussion

Increasing model capacity via increasing cardinality is more efficient than deepening a model or widening it (Xie et al., 2017; Yan et al., 2020). Compared to widening a model, increasing cardinality removes connections between hidden units and reduces both parameters and computation. Compared to deepening a model, increasing cardinality allows to parallelize the computation of all transformations of a set, accelerating both training and decoding.

4 Experiments

4.1 Settings

We conducted our experiments on the challenging massively many-to-many translation task on the OPUS-100 corpus (Tiedemann, 2012; Aharoni et al., 2019; Zhang et al., 2020). We followed Zhang et al. (2020) for experiment settings. We implemented our approaches based on the Neutron implementation (Xu and Liu, 2019) of the Transformer translation model. Parameters were initialized under the Lipschitz constraint (Xu et al., 2020). We adopted BLEU (Papineni et al., 2002) for translation evaluation with the SacreBLEU toolkit (Post, 2018).[1] We report average BLEU over 94 language pairs BLEU_{94}, win ratio WR (%) compared

[1] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.4.1

Models	Direction	$BLEU_{94}$	WR	$BLEU_4$	$BLEU_{zero}$	Speed-Up
Zhang et al. (2020)	En→xx	23.36	-	19.45	14.08	1.00
	xx→En	30.98		26.78		
Ours	En→xx	**24.17**	78.72	**20.08**	**14.71**	**1.31**
	xx→En	**32.19**	87.23	**27.92**		

Table 1: Main results

Models	$BLEU_{94}$	
	En→xx	xx→En
Full	**24.17**	**32.19**
-MIMO	23.78	31.61
-MIMO-Task Attention	23.54	31.27

Table 2: Ablation on the MIMO and task-aware attention.

#Layers	#Trans.	$BLEU_{94}$	
		En→xx	xx→En
4	6	23.92	31.76
6	4	**24.17**	**32.19**
8	3	24.08	31.94

Table 3: Results of different configurations.

to Zhang et al. (2020), average BLEU over 4 selected typologically different target languages (de, zh, br, te) $BLEU_4$, and average BLEU for zero-shot translation $BLEU_{zero}$.

4.2 Main Results

For fair comparison, we use a 6-layer model where each attention/FFN block contains 4 transformations, which leads to a similar number of parameters compared to the 24-layer model of Zhang et al. (2020). Results are shown in Table 1.

Table 1 shows that our approach achieves better performance in all evaluations while being 1.31 times as fast.

4.3 Ablation Study

We study removing MIMO transformations and task-aware attention. Results are shown in Table 2.

Table 2 verifies that both mechanisms contribute to the performance.

We also examine different combinations of depth and cardinality. Results are shown in Table 3.

Table 3 shows that using 6 layers with 4 transformations in each block leads to the best performance.

Main	en	de	fr	ar	zh	ru
1	rw	sv	pt	he	ja	sh
2	yi	da	it	mt	ko	lt
3	gd	nn	ca	fa	th	sr
4	de	nb	es	ga	vi	mk
5	xh	no	mt	yo	bn	lv

Table 4: Languages with similar task-aware attention weights.

4.4 Task-Aware Attention Weight Analysis

To verify whether task-aware attention learns to aggregate similar languages together, we extract the learned task-aware attention probabilities, flatten them into vectors, and select the languages with the top-5 cosine similarity. Results for several languages are shown in Table 4.

Table 4 shows that close languages are aggregated together.

5 Related Work

Multilingual NMT includes one-to-many (Dong et al., 2015), many-to-many (Firat et al., 2016a) and zero-shot (Firat et al., 2016b) scenarios. A simple solution is to insert a target language token at the beginning of the input sentence (Johnson et al., 2017).

Multilingual NMT has to handle different languages in one joint representation space, neglecting their linguistic diversity, especially for massively multilingual NMT (Aharoni et al., 2019; Zhang et al., 2020; Freitag and Firat, 2020). Most studies focus on how to mitigate this representation bottleneck (Zoph and Knight, 2016; Blackwood et al., 2018; Wang et al., 2018; Platanios et al., 2018; Wang et al., 2019a; Tan et al., 2019b; Wang et al., 2019b; Tan et al., 2019a; Bapna and Firat, 2019; Zhu et al., 2020; Lyu et al., 2020).

There are also studies on the trade-off between

shared and language-specific parameters (Sachan and Neubig, 2018; Zhang et al., 2021), on the training of multilingual NMT (Al-Shedivat and Parikh, 2019; Siddhant et al., 2020; Wang et al., 2020b,a), and on analyzing translations from multilingual NMT (Lakew et al., 2018) or the trained model (Kudugunta et al., 2019; Oncevay et al., 2020). Transferring a pre-trained multilingual NMT model can help improve the performance of downstream language pairs (Kim et al., 2019; Lin et al., 2020), especially for low-resource scenarios (Dabre et al., 2019). Multilingual data also has been proven useful for unsupervised NMT (Sen et al., 2019; Sun et al., 2020).

6 Conclusion

We propose to efficiently increase the capacity for multilingual NMT by increasing the cardinality. We present a MIMO architecture that allows each transformation of the block to have its own input. We also present a task-aware attention mechanism to learn to selectively utilize individual transformations from a set of transformations for different translation directions.

Our model surpasses previous work and establishes a new state-of-the-art on the large scale OPUS-100 corpus while being 1.31 times as fast.

Acknowledgments

We thank anonymous reviewers for their insightful comments. Hongfei Xu acknowledges the support of China Scholarship Council ([2018]3101, 201807040056). Josef van Genabith is supported by the German Federal Ministry of Education and Research (BMBF) under funding code 01IW20010 (CORA4NLP). Deyi Xiong is partially supported by the joint research center between GTCOM and Tianjin University and the Royal Society (London) (NAF\R1\180122).

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1184–1197, Minneapolis, Minnesota. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Graeme Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3112–3122, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416, Hong Kong, China. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine

translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257, Florence, Italy. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 641–652, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Sungwon Lyu, Bokyung Son, Kichang Yang, and Jaekyoung Bae. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5905–5918, Online. Association for Computational Linguistics.

Arturo Oncevay, Barry Haddow, and Alexandra Birch. 2020. Bridging linguistic typology and multilingual machine translation with multi-view language representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2391–2406, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 425–435, Brussels, Belgium. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Brussels, Belgium. Association for Computational Linguistics.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multilingual unsupervised NMT using shared encoder and language-specific decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3083–3089, Florence, Italy. Association for Computational Linguistics.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 2827–2835, Online. Association for Computational Linguistics.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535, Online. Association for Computational Linguistics.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019a. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019b. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019a. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020a. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

Yining Wang, Long Zhou, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2019b. A compact and language-sensitive multilingual translation method. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1213–1223, Florence, Italy. Association for Computational Linguistics.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020b. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034, Online. Association for Computational Linguistics.

Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hongfei Xu and Qiuhui Liu. 2019. Neutron: An Implementation of the Transformer Translation Model and its Variants. *arXiv preprint arXiv:1903.07402*.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. 2020. Lipschitz constrained parameter initialization for deep transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 397–402, Online. Association for Computational Linguistics.

Jianhao Yan, Fandong Meng, and Jie Zhou. 2020. Multi-unit transformers for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1047–1059, Online. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

Adaptive Nearest Neighbor Machine Translation

Xin Zheng[1], **Zhirui Zhang**[2], **Junliang Guo**[3], **Shujian Huang**[1]*, **Boxing Chen**[2], **Weihua Luo**[2] **and Jiajun Chen**[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Machine Intelligence Technology Lab, Alibaba DAMO Academy
[3]University of Science and Technology of China
[1]zhengxin@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn
[2]{zhirui.zzr, boxing.cbx, weihua.luowh}@alibaba-inc.com
[3]guojunll@mail.ustc.edu.cn

Abstract

kNN-MT, recently proposed by Khandelwal et al. (2020a), successfully combines pre-trained neural machine translation (NMT) model with token-level k-nearest-neighbor (kNN) retrieval to improve the translation accuracy. However, the traditional kNN algorithm used in kNN-MT simply retrieves a same number of nearest neighbors for each target token, which may cause prediction errors when the retrieved neighbors include noises. In this paper, we propose Adaptive kNN-MT to dynamically determine the number of k for each target token. We achieve this by introducing a light-weight *Meta-k Network*, which can be efficiently trained with only a few training samples. On four benchmark machine translation datasets, we demonstrate that the proposed method is able to effectively filter out the noises in retrieval results and significantly outperforms the vanilla kNN-MT model. Even more noteworthy is that the Meta-k Network learned on one domain could be directly applied to other domains and obtain consistent improvements, illustrating the generality of our method. Our implementation is open-sourced at https://github.com/zhengxxn/adaptive-knn-mt.

1 Introduction

Retrieval-based methods (Gu et al., 2018; Zhang et al., 2018; Bapna and Firat, 2019; Khandelwal et al., 2020a) are increasingly receiving attentions from the machine translation (MT) community recently. These approaches complement advanced neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017; Hassan et al., 2018) to alleviate the performance degradation when translating out-of-domain sentences (Dou et al., 2019; Wei et al., 2020), rare words (Koehn and Knowles, 2017), etc. The ability of accessing any provided datastore during translation makes them scalable, adaptable and interpretable.

kNN-MT, recently proposed in (Khandelwal et al., 2020a), equips a pre-trained NMT model with a kNN classifier over a datastore of cached context representations and corresponding target tokens, providing a simple yet effective strategy to utilize cached contextual information in inference. However, the hyper-parameter k is fixed for all cases, which raises some potential problems. Intuitively, the retrieved neighbors may include noises when the target token is relatively hard to determine (e.g., relevant context is not enough in the datastore). And empirically, we find that the translation quality is very sensitive to the choice of k, results in the poor robustness and generalization performance.

To tackle this problem, we propose Adaptive kNN-MT that determines the choice of k regarding each target token adaptively. Specifically, instead of utilizing a fixed k, we consider a set of possible k that are smaller than an upper bound K. Then, given the retrieval results of the current target token, we propose a light-weight *Meta-k Network* to estimate the importance of all possible k-Nearest Neighbor results, based on which they are aggregated to obtain the final decision of the model. In this way, our method dynamically evaluate and utilize the neighbor information conditioned on different target tokens, therefore improve the translation performance of the model.

We conduct experiments on multi-domain machine translation datasets. Across four domains, our approach can achieve 1.44$\sim$2.97 BLEU score improvements over the vanilla kNN-MT on average when $K \geq 4$. The introduced light-weight Meta-k Network only requires thousands of parameters and can be easily trained with a few training samples. In addition, we find that the Meta-k Net-

* Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 368–374
August 1–6, 2021. ©2021 Association for Computational Linguistics

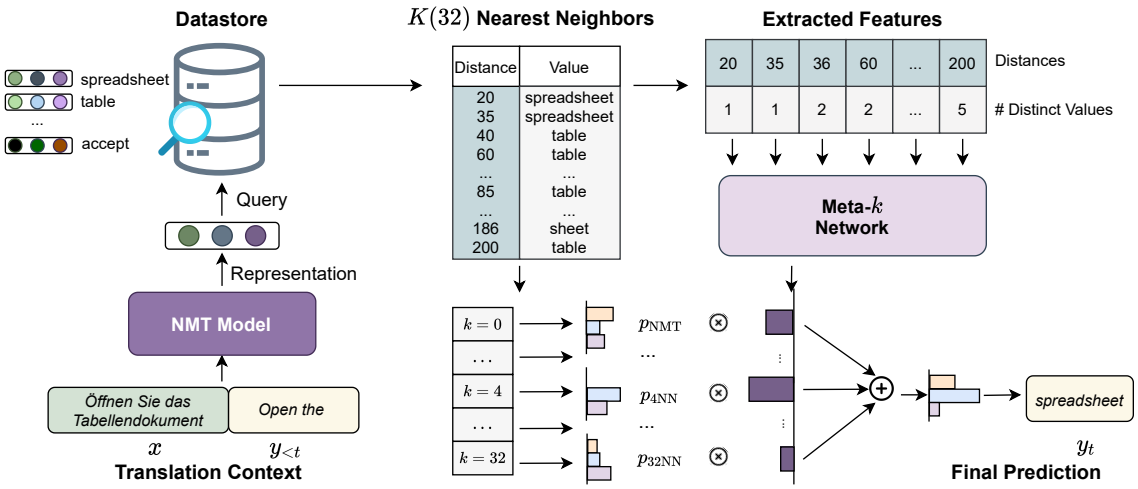


Figure 1: An overview of the proposed Adaptive kNN-MT, which could dynamically evaluate and aggregate a set of kNN predictions based on the distances as well as count of distinct values of retrieved neighbors.

work trained on one domain can be directly applied to other domains and obtain strong performance, showing the generality and robustness of the proposed method.

2 Background: kNN-MT

In this section, we will briefly introduce the background of kNN-MT, which includes two steps: creating a datastore and making predictions depends on it.

Datastore Creation. The datastore consists of a set of key-value pairs. Formally, given a bilingual sentence pair in the training set $(x, y) \in (\mathcal{X}, \mathcal{Y})$, a pre-trained autoregressive NMT decoder translates the t-th target token y_t based on the translation context $(x, y_{<t})$. Denote the hidden representations of translation contexts as $f(x, y_{<t})$, then the datastore is constructed by taking $f(x, y_{<t})$ as keys and y_t as values,

$$(\mathcal{K}, \mathcal{V}) = \bigcup_{(x,y)\in(\mathcal{X},\mathcal{Y})} \{(f(x, y_{<t}), y_t), \forall y_t \in y\}.$$

Therefore, the datastore can be created through a single forward pass over the training set $(\mathcal{X}, \mathcal{Y})$.

Prediction. While inference, at each decoding step t, the kNN-MT model aims to predict $\hat{y}_t$ given the already generated tokens $\hat{y}_{<t}$ as well as the context representation $f(x, \hat{y}_{<t})$, which is utilized to query the datastore for k nearest neighbors w.r.t the l_2 distance. Denote the retrieved neighbors as $N^t = \{(h_i, v_i), i \in \{1, 2, ..., k\}\}$, their distribution over the vocabulary is computed as:

$$p_{\text{kNN}}(y_t|x, \hat{y}_{<t}) \propto \sum_{(h_i,v_i)} \mathbb{1}_{y_t=v_i} \exp\left(\frac{-d(h_i, f(x, \hat{y}_{<t}))}{T}\right), \quad (1)$$

where T is the temperature and $d(\cdot, \cdot)$ indicates the l_2 distance. The final probability when predicting y_t is calculated as the interpolation of two distributions with a hyper-parameter λ:

$$p(y_t|x, \hat{y}_{<t}) = \lambda\, p_{\text{kNN}}(y_t|x, \hat{y}_{<t}) + (1 - \lambda)\, p_{\text{NMT}}(y_t|x, \hat{y}_{<t}), \quad (2)$$

where p_{NMT} indicates the vanilla NMT prediction.

3 Adaptive kNN-MT

The vanilla kNN-MT method utilizes a fixed number of translation contexts for every target token, which fails to exclude noises contained in retrieved neighbors when there are not enough relevant items in the datastore. We show an example with $k = 32$ in Figure 1. The correct prediction *spreadsheet* has been retrieved as top candidates. However, the model will finally predict *table* instead because it appears more frequently in the datastore than the correct prediction. A naive way to filter the noises is to use a small k, but this will also cause over-fitting problems for other cases. In fact, the optimal choice of k varies when utilizing different datastores in vanilla kNN-MT, leading to poor robustness and generalizability of the method, which is empirically discussed in Section 4.2.

To tackle this problem, we propose a dynamic method that allows each untranslated token to utilize different numbers of neighbors. Specifically,

we consider a set of possible ks that are smaller than an upper bound K, and introduce a light-weight *Meta-k Network* to estimate the importance of utilizing different ks. Practically, we consider the powers of 2 as the choices of k for simplicity, as well as $k = 0$ which indicates ignoring kNN and only utilizing the NMT model, i.e., $k \in \mathcal{S}$ where $\mathcal{S} = \{0\} \cup \{k_i \in \mathbb{N} \mid \log_2 k_i \in \mathbb{N}, k_i \leq K\}$. Then the Meta-$k$ Network evaluates the probability of different kNN results by taking retrieved neighbors as inputs.

Concretely, at the t-th decoding step, we first retrieve K neighbors N^t from the datastore, and for each neighbor (h_i, v_i), we calculate its distance from the current context representation $d_i = d(h_i, f(x, \hat{y}_{<t}))$, as well as the count of distinct values in top i neighbors c_i. Denote $d = (d_1, ..., d_K)$ as distances and $c = (c_1, ..., c_K)$ as counts of values for all retrieved neighbors, we then concatenate them as the input features to the Meta-k Network. The reasons of doing so are two-fold. Intuitively, the distance of each neighbor is the most direct evidence when evaluating their importance. In addition, the value distribution of retrieved results is also crucial for making the decision, i.e., if the values of each retrieved results are distinct, then the kNN predictions are less credible and we should depend more on NMT predictions.

We construct the Meta-k Network $f_{\text{Meta}}(\cdot)$ as two feed-forward Networks with non-linearity between them. Given $[d; c]$ as input, the probability of applying each kNN results is computed as:

$$p_{\text{Meta}}(k) = \text{softmax}(f_{\text{Meta}}([d; c])). \quad (3)$$

Prediction. Instead of introducing the hyper-parameter λ as Equation (2), we aggregate the NMT model and different kNN predictions with the output of the Meta-k Network to obtain the final prediction:

$$p(y_t|x, \hat{y}_{<t}) = \sum_{k_i \in \mathcal{S}} p_{\text{Meta}}(k_i) \cdot p_{k_i\text{NN}}(y_t|x, \hat{y}_{<t}), \quad (4)$$

where $p_{k_i\text{NN}}$ indicates the k_i Nearest Neighbor prediction results calculated as Equation (1).

Training. We fix the pre-trained NMT model and only optimize the Meta-k Network by minimizing the cross entropy loss following Equation (4), which could be very efficient by only utilizing hundreds of training samples.

4 Experiments

4.1 Experimental Setup

We evaluate the proposed model in domain adaptation machine translation tasks, in which a pre-trained general-domain NMT model is used to translate domain-specific sentences with kNN searching over an in-domain datastore. This is the most appealing application of kNN-MT as it could achieve comparable results with an in-domain NMT model but without training on any in-domain data. We denote the proposed model as Adaptive kNN-MT (A) and compare it with two baselines. One of that is vanilla kNN-MT (V) and the other is uniform kNN-MT (U) where we set equal confidence for each kNN prediction.

Datasets and Evaluation Metric. We use the same multi-domain dataset as the baseline (Khandelwal et al., 2020a), and consider domains including **IT**, **Medical**, **Koran**, and **Law** in our experiments. The sentence statistics of datasets are illustrated in Table 1. The Moses toolkit[1] is used to tokenize the sentences and split the words into subword units (Sennrich et al., 2016) with the bpe-codes provided by Ng et al. (2019). We use SacreBLEU[2] to measure all results with case-sensitive detokenized BLEU (Papineni et al., 2002).

Dataset	IT	Medical	Koran	Laws
Train	222,927	248,009	17,982	467,309
Dev	2000	2000	2000	2000
Test	2000	2000	2000	2000

Table 1: Statistics of dataset in different domains.

Implementation Details. We adopt the fairseq toolkit[3](Ott et al., 2019) and faiss[4](Johnson et al., 2017) to replicate kNN-MT and implement our model. We apply the WMT'19 German-English news translation task winner model (Ng et al., 2019) as the pre-trained NMT model which is also used by Khandelwal et al. (2020a). For kNN-MT, we carefully tune the hyper-parameter λ in Equation (2) and report the best scores for each domain. More details are included in the supplementary materials. For our method, the hidden size of the two-layer FFN in Meta-k Network is set to 32. We

[1] https://github.com/moses-smt/mosesdecoder

[2] https://github.com/mjpost/sacrebleu

[3] https://github.com/pytorch/fairseq

[4] https://github.com/facebookresearch/faiss

	Domain	IT (Base NMT: 38.35)			Med (Base NMT: 39.99)			Koran (Base NMT: 16.26)			Law (Base NMT: 45.48)			Avg (Base NMT: 35.02)		
	Model	V	U	A	V	U	A	V	U	A	V	U	A	V	U	A
K	1	42.19	41.21	42.52	51.41	50.32	51.82	18.12	17.15	18.10	58.76	58.05	58.81	42.62	41.68	42.81
	2	44.20	41.43	46.18	53.65	52.44	55.20	19.37	17.36	19.12	60.80	59.81	61.76	44.50	42.76	45.56
	4	44.89	42.31	47.23	54.16	53.01	55.84	19.50	17.88	19.69	61.31	60.75	62.89	44.97	43.49	46.41
	8	45.96	42.46	**48.04**	54.06	53.46	56.31	20.12	18.59	20.57	61.12	61.37	**63.21**	45.32	43.97	47.03
	16	45.36	43.05	47.71	53.54	54.08	**56.41**	20.30	19.45	**21.09**	60.21	61.52	63.07	44.85	44.53	**47.07**
	32	44.81	43.78	47.68	52.52	53.95	56.21	19.66	19.99	20.96	59.04	61.53	63.03	44.00	44.81	46.97
	$\sigma^2{}_{(K \geq 4)}$	0.21	0.33	**0.08**	0.42	0.18	**0.05**	**0.10**	0.65	0.30	0.81	0.10	**0.01**	0.24	0.26	**0.07**

Table 2: The BLEU scores of the vanilla kNN-MT (V) and uniform kNN-MT (U) baselines and the proposed Adaptive kNN-MT model (A). Underline results indicate the best results of baselines, and our best results are marked bold. σ^2 indicates the variance of results among different Ks.

Adaptive kNN-MT	IT	Medical	Koran	Law	Avg
In-domain	47.68	56.21	20.96	63.03	46.97
IT domain	47.68	56.20	20.52	62.33	46.68

Table 3: Generality Evaluation. We train the model on the IT domain and directly apply to other test sets.

Model	IT ⇒ Medical	Medical ⇒ IT
Base NMT	39.99	38.35
kNN-MT	25.82	15.79
Adaptive kNN-MT	37.78	30.09

Table 4: Robustness Evaluation, where the test sets are from Medical/IT domains and the datastore are from IT/Medical domains respectively.

directly use the dev set (about 2k sents) to train the Meta-k Network for about 5k steps. We use Adam (Kingma and Ba, 2015) to optimize our model, the learning rate is set to 3e-4 and batch size is set to 32 sentences.

4.2 Main Results

The experimental results are listed in Table 2. We can observe that the proposed Adaptive kNN-MT significantly outperforms the vanilla kNN-MT on all domains, illustrating the benefits of dynamically determining and utilizing the neighbor information for each target token. In addition, the performance of the vanilla model is sensitive to the choice of K, while our proposed model is more robust with smaller variance. More specifically, our model achieves better results when choosing larger number of neighbors, while the vanilla model suffers from the performance degradation when $K = 32$, indicating that the proposed Meta-k Network is able to effectively evaluate and filter the noise in retrieved neighbors, while a fixed K cannot. We also compare our proposed method with another naive baseline, uniform kNN-MT, where we set equal confidence for each kNN prediction and make it close to the vanilla kNN-MT with small k. It further demonstrates that our method could really learn something useful but not bias smaller k.

Generality. To demonstrate the generality of our method, we directly utilize the Meta-k Network trained on the IT domain to evaluate other domains. For example, we use the Meta-k Network trained on IT domain and medical datastore to evaluate the performance on medical test set. For comparison, we collect the in-domain results from Table 2. We set $K = 32$ for both settings. As shown in Table 3, the Meta-k Network trained on the IT domain achieves comparable performance on all other domains which re-train the Meta-k Network with in-domain dataset. These results also indicate that the mapping from our designed feature to the confidence of retrieved neighbors is common across different domains.

Robustness. We also evaluate the robustness of our method in the domain-mismatch setting, where we consider a scenario that the user inputs an out-of-domain sentence (e.g. IT domain) to a domain-specific translation system (e.g. medical domain) to evaluate the robustness of different methods. Specifically, in IT ⇒ Medical setting, we firstly use medical dev set and datastore to tune hyper-parameter for vanilla kNN-MT or train the Meta-k Network for Adaptive kNN-MT, and then use IT test set to test the model with medical datastore. We set $K = 32$ in this experiment. As shown in Table 4, the retrieved results are highly noisy so that the vanilla kNN-MT encounters drastic performance degradation. In contrast, our method could effectively filter out noises and therefore prevent performance degradation as much as possible.

Case Study. Table 5 shows a translation example selected from the test set in **Medical** domain with

Source	Wenn eine gleichzeitige Behandlung mit Vitamin K Antagonisten erforderlich ist, müssen die Angaben in Abschnitt 4.5 beachtet werden.
Reference	therapy with vitamin K antagonist should be administered in accordance with the information of Section 4.5.
Base NMT	If a simultaneous treatment with vitamin K antagonists is required, the information in section 4.5 must be observed.
kNN-MT	If concomitant treatment with vitamin K antagonists is required, please refer to section 4.5.
Adaptive kNN-MT	When required, concomitant ***therapy with vitamin K antagonist should be administered in accordance with the information of Section 4.5.***

Table 5: Translation examples of different systems in Medical domain.

Adaptive kNN-MT ($K = 8$)	48.04
- value count feature	46.76
- distance feature	45.60

Table 6: Effect of different features in Meta-k Network.

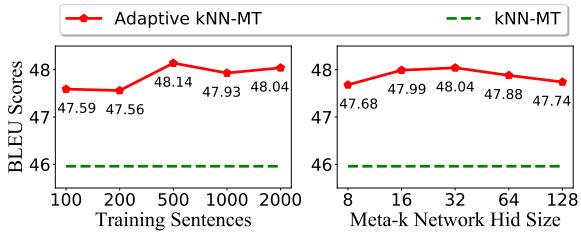


Figure 2: Effect of different number of training sentences and the hidden size of Meta-k Network.

$K = 32$. We can observe that the Meta-k Network could determine the choice of k for each target token respectively, based on which Adaptive kNN-MT leverages in-domain datastore better to achieve proper word selection and language style.

Analysis. Finally, we study the effect of two designed features, number of training sentences and the hidden size of the proposed Meta-k Network. We conduct these ablation study on IT domain with $K = 8$. All experimental results are summarized in Table 6 and Figure 2. It's obvious that both of the two features contribute significantly to the excellent performance of our model, in which the distance feature is more important. And surprisingly, our model could outperforms the vanilla kNN-MT with only 100 training sentences, or with a hidden size of 8 that only contains around 0.6k parameters, showing the efficiency of our model.

5 Conclusion and Future Works

In this paper, we propose Adaptive kNN-MT model to dynamically determine the utilization of retrieved neighbors for each target token, by introducing a light-weight *Meta-k Network*. In the experiments, on the domain adaptation machine translation tasks, we demonstrate that our model is able to effectively filter the noises in retrieved neighbors and significantly outperform the vanilla kNN-MT baseline. In addition, the superiority of our method on generality and robustness is also verified. In the future, we plan to extend our method to other tasks like Language Modeling, Question Answering, etc, which can also benefit from utilizing kNN searching (Khandelwal et al., 2020b; Kassner and Schütze, 2020).

6 Acknowledgments

We would like to thank the anonymous reviewers for the helpful comments. This work was supported by the National Key R&D Program of China (No. 2019QY1806), National Science Foundation of China (No. 61772261, U1836221) and Alibaba Group through Alibaba Innovative Research Program. We appreciate Weizhi Wang, Hao-Ran Wei and Jun Xie for the fruitful discussions. The work was done when the first author was an intern at Alibaba Group.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou, Junjie Hu, Antonios Anastasopoulos, and Graham Neubig. 2019. Unsupervised domain adaptation for neural machine translation with domain-

aware feature embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1417–1422, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5133–5140. AAAI Press.

Hany Hassan, Anthony Aue, C. Chen, Vishal Chowdhary, J. Clark, C. Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, W. Lewis, M. Li, Shujie Liu, T. Liu, Renqian Luo, Arul Menezes, Tao Qin, F. Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and M. Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *ArXiv*, abs/1803.05567.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN search component to pretrained language models for better QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3424–3430, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020a. Nearest neighbor machine translation. *CoRR*, abs/2010.00710.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020b. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hao-Ran Wei, Zhirui Zhang, Boxing Chen, and Weihua Luo. 2020. Iterative domain-repaired back-translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5884–5893, Online. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

Datastore	IT	Medical	Koran	Laws
Size	3,613,350	6,903,320	524,400	19,070,000
Hard Disk Space (Datastore)	6.9 Gb	15 Gb	1.1 Gb	37 Gb
Hard Disk Space (faiss index)	266 Mb	492 Mb	54 Mb	1.3 Gb

Table 7: Statistics of datastore in different domains.

ms / sent	K	Batch=1	Batch=16	Batch=32	Batch=64
NMT	0	165	16.3	10.5	7.9
kNN-MT	8	291.0(×1.76)	51.0(×3.1)	43.6(×4.2)	38.0(×4.8)
	16	311.4(×1.89)	81.1(×5.0)	70.4(×6.7)	64.4(×8.2)
	32	385.5(×2.34)	136.5(×8.4)	123.8(×11.8)	114.9(×14.5)
Adaptive kNN-MT	8	299.1(×1.81)	51.1(×3.1)	42.8(×4.1)	38.1(×4.8)
	16	315.0(×1.91)	80.2(×4.9)	70.2(×6.7)	63.7(×8.1)
	32	394.5(×2.40)	147.5(×9.0)	128.0(×12.2)	116.8(×14.8)

Table 8: Decoding time of different models. All results are tested on 20 cores Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz with a V100-32GB GPU.

A Appendix

A.1 Datastore Creation

We first use numpy array to save the key-value pairs over training sets as datastore. Then, faiss is used to build index for each datastore to carry out fast nearest neighbor search. We utilize faiss to learn $4k$ cluster centroids for each domain, and search 32 clusters for each target token in decoding. The size of datastore (count of target tokens), and hard disk space of datastore as well as faiss index are shown in Table 7.

A.2 Hyper-Parameter Tuning for kNN-MT

The performance of vanilla kNN-MT is highly related to the choice of hyper-parameter, i.e. k, T and λ. We fix T as 10 for IT, Medical, Law, and 100 for Koran in all experiments. Then, we tuned k and λ for each domain when using kNN-MT and the optimal choice for each domain are shown in Table 9. The performance of kNN-MT is unstable with different hyper-parameters while our Adaptive kNN-MT avoids this problem.

Dataset	IT	Medical	Koran	Laws
k	8	4	16	4
T	10	10	10	100
λ	0.7	0.8	0.8	0.8

Table 9: Optimal choice of hyper-parameters for each domain in vanilla kNN-MT.

A.3 Decoding Time

We compare the decoding time on IT test set of NMT, kNN-MT (our replicated) and Adaptive kNN-MT condition on different batch size. In decoding, the beam size is set to 4 with length penalty 0.6. The results are summarized in Table 8.

On Orthogonality Constraints for Transformers

Aston Zhang[‡,*], **Alvin Chan**[◇,*], **Yi Tay**[†], **Jie Fu**[◁], **Shuohang Wang**[○], **Shuai Zhang**[•], **Huajie Shao**[▷], **Shuochao Yao**[⋆], **Roy Ka-Wei Lee**[^]
[‡]AWS, [◇]NTU Singapore, [†]Google, [◁]Mila, Université de Montréal
[○]SMU, [•]ETH Zürich, [▷]UIUC, [⋆]George Mason University, [^]SUTD
az@astonzhang.com

Abstract

Orthogonality constraints encourage matrices to be orthogonal for numerical stability. These plug-and-play constraints, which can be conveniently incorporated into model training, have been studied for popular architectures in natural language processing, such as convolutional neural networks and recurrent neural networks. However, a dedicated study on such constraints for transformers has been absent. To fill this gap, this paper studies orthogonality constraints for transformers, showing the effectiveness with empirical evidence from ten machine translation tasks and two dialogue generation tasks. For example, on the large-scale WMT'16 En→De benchmark, simply plugging-and-playing orthogonality constraints on the original transformer model (Vaswani et al., 2017) increases the BLEU from 28.4 to 29.6, coming close to the 29.7 BLEU achieved by the very competitive dynamic convolution (Wu et al., 2019).

1 Introduction

Transformers (Vaswani et al., 2017) are a class of neural architectures that have made a tremendous transformative impact on modern natural language processing research and applications. Transformers have not only served as a powerful inductive bias for general-purpose sequence transduction (Ott et al., 2018) but also lived as the core of large pre-trained language models (Devlin et al., 2018; Radford et al., 2018; Dai et al., 2019). That said, the study of more effective training for this class of models is still an open research question, bearing great potential to impact a myriad of applications and domains.

To improve numerical stability during training, the trick of enforcing orthogonality constraints has surfaced recently. In the analysis of numerical stability, enforcing orthogonality constraints can upper-bound the Lipschitz constant of linear transformations. The Lipschitz constant is a measure that approximates the rate of change (variation) of representations. Theoretically, controlling the Lipschitz constant, which may be achieved via orthogonality constraints, yields representations that are robust and less sensitive to perturbations.

In view of this, orthogonality constraints have been studied for convolutional neural networks (CNNs) (Bansal et al., 2018; Huang et al., 2018) and recurrent neural networks (RNNs) (Arjovsky et al., 2016; Vorontsov et al., 2017; Rodríguez et al., 2016). Such plug-and-play constraints can be incorporated into model training without additional hassle. For example, CNN-based models incorporating orthogonality constraints have demonstrated empirical effectiveness for tasks such as person re-identification (Han et al., 2019) and keyword spotting (Lee et al., 2019), while RNN-based models that enforce such constraints have shown promising empirical results for response generation (Tao et al., 2018) and text classification (Wei et al., 2020; Krishnan et al., 2020). However, a dedicated study on orthogonality constraints for transformers has been absent so far.

To fill this research gap, we study orthogonality constraints for transformers, which are imposed on (i) linear transformations in self-attention and position-wise feed-forward networks and (ii) the affinity matrix in self-attention. Mathematically, orthogonality constraints on the weights of these linear transformations can be motivated by bounded Lipschitz constants. We also formally analyze the self-attention mechanism by bounding perturbations to the affinity matrix in the face of input changes.

Furthermore, we conduct extensive experiments on ten neural machine translation (both subword-

[*]Equal contribution.
[†]Work was done at NTU.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 375–382
August 1–6, 2021. ©2021 Association for Computational Linguistics

level and character-level) tasks and two dialogue generation tasks. Our experimental results are promising, demonstrating that the performance of transformers can be consistently boosted with orthogonality constraints. For example, on the large-scale WMT'16 En→De benchmark, simply plugging-and-playing orthogonality constraints on the original transformer model (Vaswani et al., 2017) increases the BLEU from 28.4 to 29.6, coming close to the 29.7 BLEU achieved by the very competitive dynamic convolution (Wu et al., 2019).

Notation For any vector $\mathbf{x}$ and any matrix $\mathbf{X}$, $\|\mathbf{x}\|$ and $\|\mathbf{X}\|$ denote their L_2-norm and spectral norm, respectively.

2 Orthogonality Constraints for Transformers

Recall that in the transformer architecture, keys, queries, and values all come from the same place in the self-attention module. They are linearly transformed for computing multiple attention heads, where all the heads are aggregated by another linear transformation. The position-wise feed-forward network is also built on two linear transformations with activations. In the following, we will describe orthogonality constraints for (i) linear transformations in self-attention and position-wise feed-forward networks and (ii) the affinity matrix in self-attention.

2.1 For Linear Transformations in Self-Attention and Position-wise Feed-Forward Networks

Note that linear transformations in self-attention and position-wise feed-forward networks are in the form:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b},$$

where $\mathbf{y}$ is the output, $\mathbf{x}$ is an input, $\mathbf{W}$ is a linear transformation weight matrix, and $\mathbf{b}$ is an optional bias term. This form provides us with convenient tools for motivating the application of orthogonality constraints to the weights of such linear transformations.

Specifically, as described in Section 1, robustness of linear transformations to small perturbations can be measured by Lipschitz constants. Thus, we begin with motivating orthogonality constraints from the perspective of bounding Lipschitz constants of linear transformations.

Formally, the linear transformation (layer) of the aforementioned form $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$ has a Lipschitz constant equal to the largest singular value of the weight matrix $\mathbf{W}$. The linear layer is Lipschitz continuous with the constant L if for all $\mathbf{x}$ and $\mathbf{x}'$, it holds that

$$\|(\mathbf{W}\mathbf{x} + \mathbf{b}) - (\mathbf{W}\mathbf{x}' + \mathbf{b})\| \leq L\|\mathbf{x} - \mathbf{x}'\|,$$

which can be re-written as

$$\frac{\|\mathbf{W}(\mathbf{x} - \mathbf{x}')\|}{\|\mathbf{x} - \mathbf{x}'\|} \leq L,$$

where $\mathbf{x} \neq \mathbf{x}'$. Therefore, the smallest Lipschitz constant is

$$\sup_{\mathbf{x} \neq \mathbf{x}'} \frac{\|\mathbf{W}(\mathbf{x} - \mathbf{x}')\|}{\|\mathbf{x} - \mathbf{x}'\|}.$$

For numerical stability, our goal is to force the Lipschitz constant to be no greater than one at every linear transformation so that their multiplication throughout compositions of transformations is also upper bounded by one. Mathematically, we need to constrain the Lipschitz constant (the largest singular value) of $\mathbf{W}$ to be no greater than one, which requires the following orthogonality constraint:

$$\mathbf{W}^\top \mathbf{W} \approx \mathbf{I}.$$

Back to the context of multi-head self-attention of transformers, denote by $\mathbf{P}$ the concatenation of the linear transformation weights for the query, key, value, and the multi-head aggregation. To impose the orthogonality constraint for these linear transformations, we add the following loss to the transformer model for every layer:

$$L_{\text{LA}} = \lambda \|\mathbf{P}^\top \mathbf{P} - \mathbf{I}\|_F^2.$$

Likewise, for position-wise feed-forward network with two linear transformation weight matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, the orthogonality constraint can be imposed with another additional loss:

$$L_{\text{LF}} = \lambda \left[\|\mathbf{M}_1^\top \mathbf{M}_1 - \mathbf{I}\|_F^2 + \|\mathbf{M}_2^\top \mathbf{M}_2 - \mathbf{I}\|_F^2 \right].$$

2.2 For the Affinity Matrix in Self-Attention

In transformers, given the query matrix $\mathbf{Q}$ and the key matrix $\mathbf{K}$ in the self-attention module, the affinity matrix

$$\mathbf{A} = \text{softmax}(\alpha \mathbf{Q}\mathbf{K}^\top), \tag{1}$$

where α is typically $\frac{1}{\sqrt{d}}$ (d is the dimension of the key and the query). Given the value matrix $\mathbf{V}$, the self-attention computes representations via the matrix multiplication $\mathbf{AV}$.

Within the context of sequence transduction, when an input word token is aligned with another semantically similar token, we would expect a small change in the behavior of the self-attention mechanism, rather than a huge change in the output. In the affinity matrix $\mathbf{A}$ as defined in (1), let $\mathbf{A}_{i,*}$ be the row vector indexed by i. Essentially, each $\mathbf{A}_{i,*}$ is a probability distribution over the tokens in the sequence that directs the alignment-based pooling operation. Intuitively, for a robust self-attention mechanism, noisy perturbations should have a limited effect on the affinity scores of the tokens.

More formally, let us analyze the self-attention mechanism by bounding perturbations to the affinity matrix in the face of input changes. Mathematically, changes to the affinity scores are bounded such that $\|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\| \leq 2\alpha\|\mathbf{K}\|\epsilon$, where $\epsilon = \|\mathbf{Q}'_{i,*} - \mathbf{Q}_{i,*}\|$ is the noise from the query matrix. We can see this as the result of the following theorem.

Theorem 2.1 (Bounded Perturbations to the Affinity Matrix)**.** *Expressing $\mathbf{A}_{i,*}$ to be the i^{th} row of the affinity matrix $\mathbf{A}$ as defined in (1) and $\mathbf{Q}_{i,*}$ to be the i^{th} row of the query matrix $\mathbf{Q}$, the perturbation to the affinity matrix is bounded as such:*

$$\|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\| \leq 2\alpha\|\mathbf{K}\|\epsilon,$$

where $\mathbf{A}' = softmax(\alpha\mathbf{Q}'\mathbf{K}^\top)$ and $\epsilon = \|\mathbf{Q}'_{i,*} - \mathbf{Q}_{i,*}\|$ is the L_2 perturbation value in $\mathbf{Q}_{i,*}$.

The detailed proof of Theorem 2.1 is provided in the appendix. In standard training, the spectral norm of the key matrix $\|\mathbf{K}\|$ or the noise ϵ from the query matrix may be large, and as a result the changes to affinity scores may become "unbounded". We speculate that this may hurt the generalization of the self-attention mechanism.

We impose orthogonality constraints on the affinity matrix $\mathbf{A}$. More concretely, we obtain an additional loss term for every layer of the transformer model using the Frobenius norm $\|\cdot\|_F$:

$$L_{\text{AM}} = \lambda\|\mathbf{A}^\top\mathbf{A} - \mathbf{I}\|_F^2,$$

where $\mathbf{I}$ is the identity matrix and λ is a scaling factor to control the ratio to the original task loss.

With orthogonally constrained affinity scores, each row vector of $\mathbf{A}$ is now orthonormal to all the other row vectors. Given that each row vector is a probability distribution over the tokens in the sequence that directs the alignment-based pooling operation, a diverse form of the self-attention mechanism would be more encouraged. This could be viewed as an additional quality of orthogonality constrained transformers.

3 Experiments

We evaluate the effectiveness of orthogonality constrained transformers (OC-transformers for brevity) on ten neural machine translation tasks and two dialogue generation tasks. Specifically, we assess three variants, largely pertaining to where orthogonality constraints are applied, i.e., (i) AM (for the affinity matrix in self-attention), (ii) LA (for the linear transformations in self-attention), and (iii) LF (for the linear transformations in position-wise feed-forward networks). We evaluate them in an incremental fashion with three main model labels: VAR-I (AM only), VAR-II (AM + LA), and VAR-III (AM + LA + LF). The scaling factor λ is tuned amongst $\{10^{-6}, 10^{-8}, 10^{-10}\}$.

3.1 Neural Machine Translation

For neural machine translation (NMT), we evaluate on both the subword-level and character-level tasks.

Experimental Setup For subword-level NMT, we evaluate our models on seven NMT datasets using the Tensor2Tensor[1] framework (Vaswani et al., 2018), namely IWSLT'14 De→En, IWSLT'14 Ro→En, IWSLT'15 En→Vi, IWSLT'17 En→Id, WMT'17 En→Et, SETIMES En→Mk, and the well-established large-scale WMT'16 En→De.

All the models are trained with the transformer-base setting. Owing to the smaller size, we use the transformer-small setting for IWSLT'14 datasets. For the WMT'16 En→De dataset, we train both the transformer-base and transformer-big settings on $4\times$ GPUs with gradient accumulation of $2\times$ to emulate $8\times$ GPU training. By determining improvement on approximate BLEU scores on the validation set, we train models for $2M$ steps for the transformer-base setting and $800K$ steps for the transformer-big setting. Note that between the standard transformer and OC-transformer, we maintain all the other hyperparameters to keep the comparisons as fair as possible. For character-level NMT, we evaluate on three language pairs, namely

[1] https://github.com/tensorflow/tensor2tensor

Table 1: Experimental results on subword-level neural machine translation.

Model	BLEU					
	De→En	Ro→En	En→Vi	En→Id	En→Et	En→Mk
Transformer	34.68	32.36	28.43	47.40	14.17	13.96
OC-transformer (VAR-I)	34.87	**32.68**	30.16	48.09	14.83	14.74
OC-transformer (VAR-II)	34.92	32.63	**30.51**	48.05	**15.06**	**14.70**
OC-transformer (VAR-III)	**35.20**	32.44	30.42	**48.33**	14.87	14.62
Relative Gain (%)	+1.5%	+1.0%	+7.3%	+2.0%	+6.3%	+5.3%

Table 2: Experimental results on neural machine translation with the WMT'16 En→De *newstest2014* test set.

Model	BLEU
MoE (Shazeer et al., 2017)	26.0
Transformer-base (Vaswani et al., 2017)	27.3
Transformer-big (Vaswani et al., 2017)	28.4
Transformer-ott-big (Ott et al., 2018)	29.3
Dynamic convolution (Wu et al., 2019)	**29.7**
OC-transformer-base based on (Vaswani et al., 2017) (VAR-III)	28.5
OC-transformer-big based on (Vaswani et al., 2017) (VAR-III)	29.6

WMT En→Fr, IWSLT'14 Ro→En, and IWSLT'15 De→En. The transformer-small setting is used for all the three language pairs and trained for $200K$ steps.

Experimental Results Table 1 reports experimental results on subword-level NMT datasets. Overall, we note that performance of transformers is consistently boosted by orthogonality constraints, ascertaining the effectiveness of adopting such plug-and-play tricks. More specifically, they are able to achieve $+1.0\%$ to $+7.3\%$ relative gain over the standard transformer. Notably, all the variants (VAR-I, VAR-II, and VAR-III) boost the performance of transformers: it demonstrates that orthogonality constraints are indeed useful. Moreover, orthogonal constraints on the self-attention affinity matrix are beneficial in general even if the rest of the model is not fully enforced with orthogonality constraints.

Table 2 reports the results on the large-scale WMT'16 En→De dataset. Orthogonality constraints boost the performance of the transformer-big setting based on (Vaswani et al., 2017), increasing the BLEU from 28.4 to 29.6. This result outperforms the more advanced transformer-ott-big proposed in (Ott et al., 2018) and comes close to 29.7 that was achieved by the very competitive dynamic convolution model (Wu et al., 2019). Likewise, orthogonality constraints also boost the performance of the transformer-base setting with the BLEU increased from 27.3 to 28.5.

Table 3 reports the results on character-level NMT. We observe that orthogonality constraints consistently boost the performance of standard transformers on all the three language pairs: En→Fr (+3.5%), Ro→En (+2.6%), and De→En (+1.6%).

3.2 Sequence-to-Sequence Dialogue Generation

We conduct experiments on the sequence-to-sequence dialog generation task whereby the goal is to generate the reply in a two-way conversation.

Experimental Setup We use two datasets: PersonaChat (Zhang et al., 2018) and DailyDialog (Li et al., 2017). We implement our task in Tensor2Tensor using the transformer-small setting in a sequence-to-sequence fashion (Sutskever et al., 2014). We train all the models for $20K$ steps, which we find sufficient for model convergence. Beam search of beam size 4 and length penalty 0.6 is adopted for decoding the output sequence. We evaluate all the models with the language generation evaluation suite in (Sharma et al., 2017).

Experimental Results Table 4 reports our results on the PersonaChat and DailyDialog datasets. The key observation is that all the variants of enforcing orthogonality constraints boost performance of standard transformers. The best results of OC-transformers make a substantial improvement in all

Table 3: Experimental results on character-level neural machine translation.

Model	BLEU		
	En→Fr	Ro→En	De→En
Transformer (Vaswani et al., 2017)	18.74	22.04	27.59
OC-transformer based on (Vaswani et al., 2017) (VAR-III)	**19.40**	**22.61**	**28.02**
Relative Gain (%)	+3.5%	+2.6%	+1.6%

Table 4: Experimental results on the PersonaChat dataset (Zhang et al., 2018) and the DailyDialog dataset (Li et al., 2017) on nine evaluation measures (Sharma et al., 2017). SkipT stands for SkipThought cosine similarity, EmbA stands for embedding average, VecE stands for vector extrema, and GreedyM stands for greedy matching.

	Transformer	OC-transformer (VAR-I)	OC-transformer (VAR-II)	OC-transformer (VAR-III)	Relative Gain
	PersonaChat				
BLEU-1	13.2	15.1	15.4	**16.3**	+23.5%
BLEU-4	2.04	2.28	2.38	**2.50**	+22.5%
Meteor	6.10	6.55	6.60	**6.70**	+9.8%
Rouge	14.2	14.7	**15.1**	**15.1**	+6.3%
CIDEr	18.2	18.7	**19.3**	18.3	+6.0%
SkipT	41.9	42.8	**43.9**	43.3	+4.8%
EmbA	84.3	84.6	**84.9**	84.6	+0.7%
VecE	**49.0**	48.2	**49.0**	48.6	+0.0%
GreedyM	65.8	66.2	**66.5**	66.4	+1.1%
	DailyDialog				
BLEU-1	12.1	13.5	13.3	**14.0**	+15.7%
BLEU-4	6.22	6.70	6.52	**7.11**	+14.3%
Meteor	8.23	8.43	8.39	**8.72**	+6.0%
Rouge	21.1	21.4	**21.7**	**21.7**	+2.8%
CIDEr	79.3	79.2	79.6	**82.1**	+3.5%
SkipT	66.9	67.1	67.1	**67.2**	+0.4%
EmbA	84.9	**85.7**	85.6	85.5	+0.9%
VecE	53.1	53.3	**53.4**	53.2	+0.5%
GreedyM	72.1	72.3	**72.6**	72.2	+0.7%

the nine evaluation measures. Notably, on both datasets, the best variants are either VAR-II or VAR-III. VAR-I performs decently and boosts performance of standard transformers on both tasks, signifying that the orthogonality constrained affinity matrix in self-attention is sufficiently effective. This mirrors the results on neural machine translation and is consistent across the findings. The relative gain of applying orthogonality constraints is also promising, peaking at $+23.5\%$ on BLEU-1 scores and $+2.8\%$ to $+6.3\%$ on Rouge.

4 Conclusion

We studied orthogonality constraints for transformers, which are imposed on (i) linear transformations in self-attention and position-wise feed-forward networks and (ii) the affinity matrix in self-attention. We showed that such plug-and-play constraints, which can be conveniently incorporated, consistently boost performance of transformers on ten different machine translation tasks and two dialogue generation tasks. For example, on the large-scale WMT'16 En→De benchmark, simply plugging-and-playing orthogonality constraints on the original transformer model (Vaswani et al., 2017) increases the BLEU from 28.4 to 29.6, coming close to the 29.7 BLEU achieved by the very competitive dynamic convolution (Wu et al., 2019).

Broader Impact Given widespread adoptions of transformer models, the proposed plug-and-play orthogonal constraints could also be useful to computer vision, automatic speech recognition, time series analysis, and biological sequence analysis.

References

Martin Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. 2018. Can we gain more from orthogonality regularizations in training deep cnns? *arXiv preprint arXiv:1810.09102*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chuchu Han, Ruochen Zheng, Changxin Gao, and Nong Sang. 2019. Complementation-reinforced attention network for person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3433–3445.

Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. 2018. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jitin Krishnan, Hemant Purohit, and Huzefa Rangwala. 2020. Diversity-based generalization for neural unsupervised text classification under domain shift. *arXiv preprint arXiv:2002.10937*.

Mingu Lee, Jinkyu Lee, Hye Jin Jang, Byeonggeun Kim, Wonil Chang, and Kyuwoong Hwang. 2019. Orthogonality constrained multi-head attention for keyword spotting. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 86–92. IEEE.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *online*.

Pau Rodríguez, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. 2016. Regularizing cnns with locally constrained decorrelations. *arXiv preprint arXiv:1611.01967*.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*, pages 4418–4424.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. 2017. On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning*, pages 3570–3578. PMLR.

Jiyao Wei, Jian Liao, Zhenfei Yang, Suge Wang, and Qiang Zhao. 2020. Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing*, 383:165–173.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

A Proof of Theorem 2.1

Proof. Let $\mathbf{x} = \mathbf{Q}_{i,*}$, $g(\mathbf{x}) = \mathbf{x}\mathbf{K}^\top$, and $f(\mathbf{y}) = \text{softmax}(\mathbf{y})$. Expressing each row in $\mathbf{A}$ as $\mathbf{A}_{i,*} = \text{softmax}(\alpha\mathbf{Q}_{i,*}\mathbf{K}^\top)$, we have

$$\mathbf{A}_{i,*} = f(\alpha g(\mathbf{x})). \tag{2}$$

We first consider bounding $g(\mathbf{x})$ with respect to $\|\mathbf{x}' - \mathbf{x}\|$:

$$\|g(\mathbf{x}') - g(\mathbf{x})\| = \|(\mathbf{x}' - \mathbf{x})\mathbf{K}^\top\|.$$

Recalling the definition of the spectral norm, $\|\mathbf{A}\| = \max_{\mathbf{x}\in\mathbb{R}^l\setminus\{\mathbf{0}\}} \frac{\|\mathbf{x}\mathbf{A}\|}{\|\mathbf{x}\|}$:

$$\|g(\mathbf{x}') - g(\mathbf{x})\| \leq \|\mathbf{K}\|\|\mathbf{x}' - \mathbf{x}\|. \tag{3}$$

Here, we can observe that the Lipschitz constant for g is $\|\mathbf{K}\|$.

Next, we bound $f(\mathbf{y}) = \text{softmax}(\mathbf{y})$ with respect to $\|\mathbf{y}' - \mathbf{y}\|$. Since f is a differentiable function, it holds that

$$\|f(\mathbf{y}') - f(\mathbf{y})\| \leq \|\mathbf{J}\|^*\|(\mathbf{y}' - \mathbf{y})\|, \tag{4}$$

where $\mathbf{J}$ is the Jacobian matrix of $f(\mathbf{y})$ with respect to $\mathbf{y}$, i.e., $\mathbf{J}_{i,j} = \frac{\partial f(\mathbf{y})_i}{\partial \mathbf{y}_j}$, and $\|\mathbf{J}\|^* = \max_{\mathbf{y}} \|\mathbf{J}\|$. Since $f(\mathbf{y})_i = \frac{e^{\mathbf{y}_i}}{\sum e^{\mathbf{y}_j}}$, for diagonal entries of $\mathbf{J}$ we have

$$\begin{aligned}\mathbf{J}_{i,i} &= \frac{\partial f(\mathbf{y})_i}{\partial \mathbf{y}_i} \\ &= \frac{e^{\mathbf{y}_i}}{\sum e^{\mathbf{y}_j}} - \frac{e^{2\mathbf{y}_i}}{(\sum e^{\mathbf{y}_j})^2} \\ &= f_i - {f_i}^2,\end{aligned}$$

where $f_i = f(\mathbf{y})_i$ for brevity. For non-diagonal entries of $\mathbf{J}$ where $i \neq j$, we have

$$\begin{aligned}\mathbf{J}_{i,j} &= \frac{\partial f(\mathbf{y})_i}{\partial \mathbf{y}_j} \\ &= -\frac{e^{\mathbf{y}_i}e^{\mathbf{y}_j}}{(\sum e^{\mathbf{y}_j})^2} \\ &= -f_i f_j.\end{aligned}$$

With this, we can express the Jacobian $\mathbf{J}$ as

$$\begin{aligned}\mathbf{J} &= \begin{bmatrix} f_1 - {f_1}^2 & \cdots & -f_1 f_n \\ \vdots & \ddots & \vdots \\ -f_n f_1 & \cdots & f_n - {f_n}^2 \end{bmatrix} \\ &= \text{diag}(f_i) - \mathbf{f}^\top\mathbf{f},\end{aligned}$$

where $\mathbf{f} = [f_1, .., f_n]$ and $\mathbf{f}^\top\mathbf{f}$ is the outer product of $\mathbf{f}$. We can then express the spectral norm of $\mathbf{J}$ as

$$\begin{aligned}\|\mathbf{J}\| &= \|\text{diag}(f_i) - \mathbf{f}^\top\mathbf{f}\| \\ &\leq \|\text{diag}(f_i)\| + \|\mathbf{f}^\top\mathbf{f}\|.\end{aligned} \tag{5}$$

Note that $\text{diag}(f_i)$ and $\mathbf{f}^\top\mathbf{f}$ are both symmetric matrices. The spectral norm of a symmetric matrix $\mathbf{M}$ is the largest absolute value of its eigenvalues λ:

$$\|\mathbf{M}\| = \max_i |\lambda_i(\mathbf{M})|. \tag{6}$$

For a diagonal matrix like $\text{diag}(f_i)$, its eigenvectors are the standard basis vector while its eigenvalues are the non-zero diagonal entries, i.e., $\lambda_i(\text{diag}(f_i)) = f_i$. Thus, we can get

$$\|\text{diag}(f_i)\| = \max_i f_i. \tag{7}$$

Next, we find $\|\mathbf{f}^\top\mathbf{f}\|$ through the eigenvalues of $\mathbf{f}^\top\mathbf{f}$. When we take the product of $\mathbf{f}^\top\mathbf{f}$ and $\mathbf{f}^\top$,

$$\begin{aligned}\mathbf{f}^\top\mathbf{f}\cdot\mathbf{f}^\top &= \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \begin{bmatrix} f_1 & \cdots & f_n \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \\ &= \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \cdot \sum_i {f_i}^2 \\ &= \left(\sum_i {f_i}^2\right)\mathbf{f}^\top.\end{aligned}$$

From this, we know $\lambda_1(\mathbf{f}^\top\mathbf{f}) = \sum_i {f_i}^2$, with the corresponding eigenvector $\mathbf{v}_1 = \mathbf{f}^\top$. Since the remaining $n-1$ eigenvectors are orthogonal to $\mathbf{v}_1$, i.e., $\mathbf{v}_1^\top\mathbf{v}_i = \mathbf{f}\mathbf{v}_i = 0, \forall i \neq 1$, we have

$$\begin{aligned}\mathbf{f}^\top\mathbf{f}\cdot\mathbf{v}_i &= \mathbf{f}^\top(\mathbf{f}\cdot\mathbf{v}_i) \\ &= \mathbf{0}.\end{aligned}$$

This implies that $\sum_i {f_i}^2$ is the only non-zero eigenvalue of $\mathbf{f}^\top\mathbf{f}$. Thus, with (6), this gives

$$\|\mathbf{f}^\top\mathbf{f}\| = \sum_i {f_i}^2.$$

Combining this with (5) and (7), we get

$$\|\mathbf{J}\| \leq \max_i f_i + \sum_i {f_i}^2. \tag{8}$$

Recall that $\|\mathbf{J}\|$ is the largest possible spectral norm of $\mathbf{J}$, i.e., $\|\mathbf{J}\|^* = \max_{\mathbf{y}} \|\mathbf{J}\|$. Moreover, by

definition of probability, it holds that $f_i \leq 1$ and sum of probabilities $\sum f_i \leq 1$. Therefore,

$$\begin{aligned} \|\mathbf{J}\|^* &\leq \max_{i,\mathbf{y}} f_i + \max_{\mathbf{y}} \sum_i {f_i}^2 \\ &\leq 1 + 1 = 2. \end{aligned} \tag{9}$$

With (4) and (9), we get

$$\|f(\mathbf{y}') - f(\mathbf{y})\| \leq 2\|(\mathbf{y}' - \mathbf{y})\|. \tag{10}$$

Bounding $\|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\|$ with (2), (10), and (3), this gives

$$\begin{aligned} \|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\| &= \|f(\alpha g(\mathbf{x}')) - f(\alpha g(\mathbf{x}))\| \\ &\leq \|2\alpha g(\mathbf{x}') - 2\alpha g(\mathbf{x})\| \\ &= 2\alpha\|g(\mathbf{x}') - g(\mathbf{x})\| \\ &\leq 2\alpha\|\mathbf{K}\|\|\mathbf{x}' - \mathbf{x}\| \\ &= 2\alpha\|\mathbf{K}\|\epsilon. \end{aligned}$$

□

Measuring and Improving BERT's Mathematical Abilities by Predicting the Order of Reasoning

Piotr Piękos
University of Warsaw
piotrpiekos@gmail.com

Mateusz Malinowski*
DeepMind

Henryk Michalewski*
University of Warsaw, Google

Abstract

Imagine you are in a supermarket. You have two bananas in your basket and want to buy four apples. How many fruits do you have in total? This seemingly straightforward question can be challenging for data-driven language models, even if trained at scale. However, we would expect such generic language models to possess some mathematical abilities in addition to typical linguistic competence. Towards this goal, we investigate if a commonly used language model, BERT, possesses such mathematical abilities and, if so, to what degree. For that, we fine-tune BERT on a popular dataset for word math problems, AQuA-RAT, and conduct several tests to understand learned representations better.

Since we teach models trained on natural language to do formal mathematics, we hypothesize that such models would benefit from training on semi-formal steps that explain how math results are derived. To better accommodate such training, we also propose new pretext tasks for learning mathematical rules. We call them (Neighbor) Reasoning Order Prediction (ROP or NROP). With this new model, we achieve significantly better outcomes than data-driven baselines and even on-par with more tailored models. We also show how to reduce positional bias in such models.

1 Introduction

Automatically solving math word problems has a long history dating back to the middle sixties (Bobrow, 1964). Early approaches were rule-based matching systems that solve the problem symbolically. Even though there are some impressive symbolic systems that operate in a relatively narrow domain, the inability to successfully scale them up is sometimes presented as a critique of the good-old-fashioned AI, or GOFAI (Dreyfus et al., 1992). One issue is to create a formalism that covers all the aspects needed to solve these problems. On the other hand, deep learning (LeCun et al., 2015) aims to develop artificial general intelligence that scales better to various problems.

However, despite many successes in computer vision and natural language processing (Devlin et al., 2018; He et al., 2016; Krizhevsky et al., 2012; Lan et al., 2019; Mikolov et al., 2013), data-driven methods evade our dream of building a system with basic, every-day, mathematical skills. As large-scale natural language models become more common (Devlin et al., 2018; Brown et al., 2020), we would expect them to also reason mathematically.

Since natural language understanding also involves symbolic manipulation (Liang, 2016), we treat mathematical reasoning as a language understanding and revisit the data-driven paradigm. For that, we rely on a recent language model, BERT (Devlin et al., 2019), and challenge it with math word problems (Ling et al., 2017). Even though such language models have initially shown promising results, more recent investigation shows they may rely on various biases in their predictions (Hendricks et al., 2018; Brown et al., 2020; Bhardwaj et al., 2020; Kurita et al., 2019). Here, we also follow that line of investigation and show these models can answer correctly without an understanding of the rationale behind it.

Furthermore, as directly predicting answers to math problems often requires multiple steps of reasoning, we show that we can improve BERT's generalization by exposing it to rationales (Ling et al., 2017; Hendricks et al., 2016; Lei et al., 2016). These are, however, only used during training similarly to a teacher that shows a student a justification for each answer. But then, the student is evaluated only on the ability to answer these questions during the college exam correctly with no access to rationales. Finally, to learn a better representation

*Authors have contributed equally.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 383–394
August 1–6, 2021. ©2021 Association for Computational Linguistics

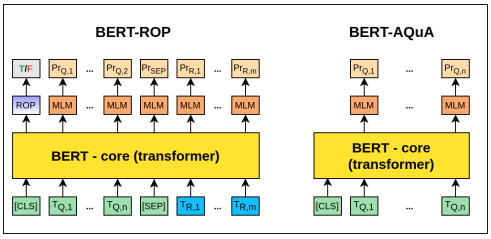


Figure 1: BERT (right) and our novel extension (left). We use shared architecture but we separate question tokens (green blocks) from rationales (blue blocks) using different segment and positional embeddings. We show all three losses. MLM predicts masked tokens (depicted here as $Pr_{Q,k}$). We use ROP or NROP to predict if the ordering of rationale steps is correct. For question-answering, we fine-tune the whole model with a classification layer using softmax. We use the embedding that corresponds to the [CLS] token as the input representation.

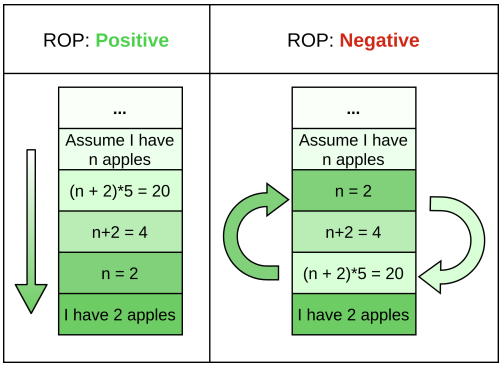


Figure 2: ROP or NROP with positive (left) and negative (right) labels. We randomly swap two rationales and classify if that change has happened.

from rationales and to improve the generalization even further, we introduce novel pretext tasks and corresponding losses, which we name (Neighbor) Reasoning Order Prediction (ROP or NROP). We also show that permutation invariant losses can lead to less biased representations. With that, we outperform other data-driven baselines, and are even on-par with methods that are more tailored to math-world problems and the AQuA-RAT dataset.

2 Methods

We use the following methods, each initialized with BERT-base pre-trained on Wikipedia and Books Corpus (Devlin et al., 2018; Zhu et al., 2015). Note that, in fine-tuning they all have the same number of parameters.

1) **BERT-base**. We fine-tune BERT to predict the correct answer and show its transfer to math word problems.

2) **BERT-AQuA**. We use the MLM loss on the AQuA-RAT questions before training to predict correct answer.

3) **BERT-AQuA-RAT**. We use the MLM loss on the AQuA-RAT questions and rationales and show if we can inject knowledge from rationales into BERT.

4) **BERT-(N)ROP**. We use the MLM loss and the novel (N)ROP loss for coherence prediction (defined later) and show if we can improve the results by focusing the model on rationales.

Later in this paper, we propose permutation invariant losses that additionally reduce positional biases of the BERT-base model, and can work with all the pretext tasks described above.

2.1 Architectures, pretext tasks and losses

We base our architecture on BERT (Devlin et al., 2019) that has 12 transformer blocks (Vaswani et al., 2017). As the core, we use the standard configuration described in (Devlin et al., 2019). We use three self-supervised losses. One is the standard Masked Language Modelling (MLM) but extended to work on rationales. Other two are our new losses, (Neighbour) Reasoning Order Prediction (ROP or NROP). Figure 1 shows two variants of our models. Note that, during fine-tuning, rationales and all the self-supervised losses are discarded.

MLM is the Masked Language Modelling (Devlin et al., 2019). We randomly mask 15% of the input tokens by a special token [MASK]. The objective of this loss is to predict the masked token using its context casted as a classification problem over the tokenizer vocabulary. Loss is calculated only on masked tokens. We extend this loss to rationales. First, we randomly choose whether we mask a question or rationale. Next, we follow the procedure above applied to either a question or rationale. However, to encourage binding between questions and rationales, we use the whole context for the predictions. Interestingly, there are parallels between masking numbers and solving mathematical equations, where it can be seen as solving the equation with unknown. For example, $2 + [\text{MASK}] = 4$ becomes $2 + x = 4$. As a consequence, models during training organically deal with mathematical calculations without defining a specific loss for mathematics allowing soft transitions between natural and more formal languages.

ROP is our novel coherence loss. Since rationales are sequences of consecutive reasoning steps, the order of the execution is critical as shown in Figure 2. Following this intuition, we introduce

Reasoning Order Prediction (ROP) that predicts whether the order of the rationale steps is preserved. Hence it encourages the network to pay more attention to rationales. The loss is similar to Sentence Order Prediction (SOP) (Lan et al., 2019), but ours is focused on learning reasoning steps.

NROP is an extension of ROP where only consecutive rationale steps are swapped making the prediction (swap or no swap) task more challenging and, hence, it can arguably lead to a better representation as understanding the correct ordering is more nuanced. Indeed, we observe that our models trained with NROP correctly predict if swap has occurred in about 75% cases, while with ROP in about 78% cases (both on the validation set). This indeed, confirms our hypothesis that NROP task is more challenging than ROP.

3 Results

Dataset. We use AQuA-RAT (Ling et al., 2017). It has about 100k crowd-sourced math questions with five candidate answers (one is correct). Each question has a rationale – a step-by-step explanation of how the answer is computed – that is only available during training. At test time answer predictions are based on questions. The train set has roughly 100k question-answer-rationale triples, while dev and test about 250 question-answer pairs each.

Main results. Table 1 shows our main results. We see that our method is the state-of-the-art among the models with minimal inductive biases and is very competitive to the other two models that are more specific to handle word math problems (e.g., requires programs). Moreover, even though BERT is already a stronger model than LSTM, it is better to use its MLM pretext task and loss on the AQuA-RAT questions (BERT-AQuA) or even better on questions and rationales (BERT-AQuA-RAT). However, models with our novel coherence prediction losses can better learn from rationales (BERT-ROP and BERT-NROP).

Moreover, we observe a highly sensitive relationship between dev and test sets (Figure 3, left), where small changes in the accuracies in the former set can lead to more dramatic changes at test time. Indeed, the correlation of results between both sets is only 0.082. As the validation set is quite small, we propose an extended dev consisting of 5000 randomly chosen samples from the training set extended by the whole dev set. Although not ideal, and the sensitive relationship is still present

Model	Accuracy
Random chance	20.0%
LSTM (Ling et al., 2017)	20.8%
BERT-base (ours)	28.3(±2.0)%
BERT-AQUA (ours)	29.1(±1.7)%
BERT-AQuA-RAT (ours)	32.3(±1.8)%
BERT-ROP (ours)	35.4(±1.0)%
BERT-NROP (ours)	37.0(±1.1)%
AQuA-RAT (Ling et al., 2017)	36.4%
MathQA (Amini et al., 2019)	**37.9%**

Table 1: Comparison of data-driven (first six rows) with two hybrid approaches that use stronger and hence more specific inductive biases (last two rows). Standard deviation estimates (over random initializations) is given in parentheses, where we see our losses can reduce the variability slightly.

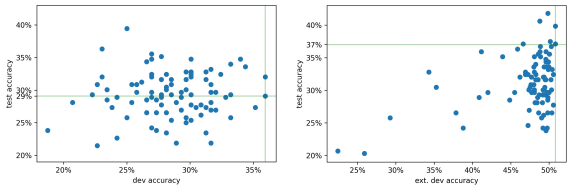


Figure 3: Accuracies for dev and test sets. Green lines show the iteration that maximizes validation accuracy. The image also shows the sensitivity of relationship between test and the original (left) or our extended (right) validation set.

(Figure 3, right), we have increased the correlation to 0.401. With such a new validation set, we report 37% test accuracy but we can also see that 40% is within the reach (Figure 3, right).

Rationales. We hypothesize that rationales contain information that is either missing or hard to extract from questions. For instance, their structure is different; they are more formal with emphasis on the logical steps. However, testing that hypothesis is non-trivial as there is a confounding factor – adding more rationales results in more data. Therefore, we artificially modify the dataset so that both models (one trained only on questions, and another one on questions and rationales) are trained on roughly the same number of data points. For that, we have estimated that rationales have 1.7 times more tokens than questions. This means that a question combined with rationale has around 3 times more tokens than just a question. If our hypothesis is valid, training on 20% questions and rationales should give better results than training on 60% questions (counting the number of tokens). We therefore created samples of respective sizes of just questions and questions combined with rationales. We show our results in Figure 4. The results suggest that adding more questions is insufficient and only slightly improves the overall performance.

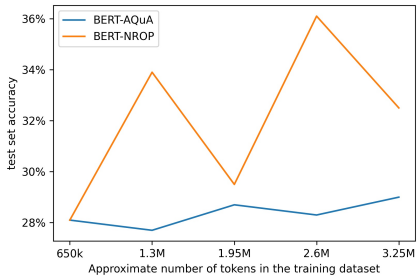


Figure 4: Accuracy scores conditioned on the number of tokens available for training. To support our argument that training on rationales is qualitatively different than questions, we align both together so that we have comparable number of tokens in both cases. Plot shows the progression of the dataset size. Starting with 650K of tokens - 20% dataset BERT-AQuA and 6.66% for BERT-NROP and ending with 3.25M - 100% of dataset for BERT-AQuA and 33.3% dataset for BERT-NROP. This shows that training with rationales leads to a better representation. Even better than training with more questions.

On the other hand, using rationales is more helpful.

Embeddings. To better understand the difference between BERT and BERT+NROP, we analyze their embeddings. For our analysis, we sample 2500 questions with a single operator in rationales, and next we visualise them with T-SNE (Van der Maaten and Hinton, 2008). We show both in Figure 5. We observe that BERT+NROP embeddings preserve more information about different operators.

Permutation consistency. Random guessing on AQuA-RAT yields 20%. With that in mind to separate questions that were solved by chance, we have constructed a new evaluation task – permutation consistency test – where each question gets 5 answers at different positions. Table 2 shows our procedure. Here, models only score a single point if they solve all 5 questions correctly. Hence, a random chance is 0.032% in such experiments.

Table 3 shows our results. BERT+NROP solves almost three times as many questions as BERT. Additionally, further inspection shows that BERT relies on choosing the answers that most stand out, e.g., numbers ending with zeros or floats while every other option is an integer. We didn't observe that simple patterns with BERT+NROP. Questions solved by BERT+NROP usually contain one or two operations and show that BERT+NROP better understands the problem. Below, we exemplify two math problems solved by both models.

Example of a problem solved by BERT+NROP: *8 man work for 6 days to complete a work. How many men are required to complete same work in 1/2 day?*
Answers: *A)93, B)94, C)95, D)96, E)97*

Original question	
How much is 27 / 3	A)13 **B)9** C)3 D)12 E)17
Generated questions	
How much is 27 / 3	**A)9** B)13 C)3 D)12 E)17
How much is 27 / 3	A)13 **B)9** C)3 D)12 E)17
How much is 27 / 3	A)13 B)3 **C)9** D)12 E)17
How much is 27 / 3	A)13 B)12 C)3 **D)9** E)17
How much is 27 / 3	A)13 B)17 C)3 D)12 **E)9**

Table 2: Our generation method for the permutation consistency test. Models get a point only if they solve all them.

Correct Option: *D*
Example of a problem solved by BERT *A ship went on a voyage. After it had traveled 180 miles a plane started with 10 times the speed of the ship. Find the distance when they meet from starting point.?*
Answers: *A)238, B)289, C)200, D)287, E)187*
Correct Option: *C*

Model	Score
Random chance	0.032%
BERT	4.33%
BERT+NROP	**11.02%**
BERT AUG	13.4%
BERT+NROP AUG	19.7%
BERT SEP-NC	15.0%
BERT+NROP SEP-NC	22.7%
BERT SEP-C	16.1%
BERT+NROP SEP-C	**23.9%**

Table 3: Our results for the permutation consistency test.

Drop from 37.0% to 11.02% (Table 3) suggests that models rely strongly on the order of answers. To reduce such a bias, we test several permutation invariant losses.

1) **AUG**. We sample randomly 25 permutations of all the possible answers and use them during training. Original ordering is not used, so there is no order bias. This is a data augmentation technique.

2) **SEP-NC**. The original models are trained on a 5-class classification task, where we build the representation by using questions and all the candidate answers, i.e., $\textbf{BERT}(Q||P)$. Here, $||$ denotes concatenation, Q is the question and P represents the sequence of all answers. In SEP-NC, we block the path between all the candidate answers and the BERT-base. Next, we use a late-fusion to predict if the given candidate answer matches with the question. That is, we use the following formulation $f(\textbf{BERT}(Q)||\textbf{BERT}(C))$, where $C \in P$ is a single candidate answer and f is a multi-layer perception (with two layers). At test time, the model

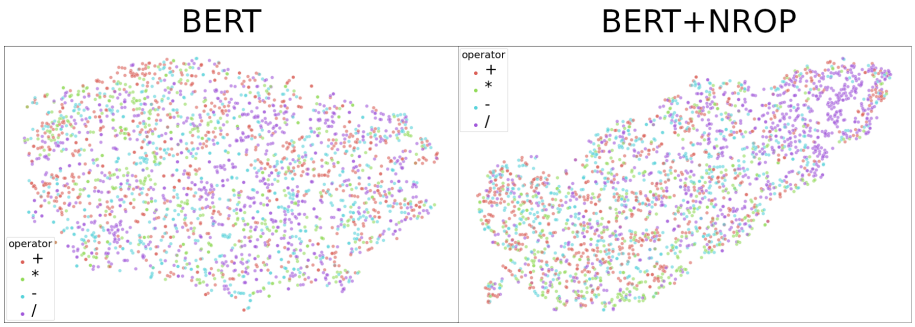


Figure 5: BERT and BERT+NROP embeddings. Colours represent different operators in rationales (T-SNE). BERT+NROP embeddings better separate operators.

is prompted to score all five candidate answers and select the one with the highest score. Appendix has more information about this method.

3) **SEP-C**. As models trained with SEP-NC do not have access to all the possible answers, their biases to answer positions are significantly reduced. However, these models cannot compare each answer to all other candidate answers. Here, we use the following formulation $f(\textbf{BERT}(Q||P)||\textbf{BERT}(C))$ to measure the compatibility of the input (question Q and all the candidate answers P) with the given candidate answer $C \in P$. We also reset the positional encoding between every possible answer in P. In such a way, we hypothesise the network can learn a less biased representation, and on the other hand, use relationship between the candidate answers. Table 3 shows SEP-NC and SEP-C vastly outperform the original model on the permutation consistency test. Details are in the appendix.

SEP-NC and SEP-C improve permutation consistency tests. Yet, they give similar results to original methods in accuracy measuring task. They achieve respectively 33.5% (SEP-NC) and 35.4% (SEP-C).

Questions difficulty. To better understand the models' performance, we check which questions are difficult for the model. We categorize questions by their difficulty for BERT-NROP and BERT. To estimate a question's difficulty, we have ranked the candidate answers according to the model's uncertainties. For instance, if the correct answer has the 2nd largest probability, we assign to that question difficulty two. With that, we group questions into 5 difficulty categories, from the easiest: $D_1, .., D_5$.

Manual inspection shows that for BERT+NROP: D_5 requires additional knowledge or implicitly defined numbers (e.g., adding first 100 numbers), D_4 requires geometry or non-linear equations and systems, D_3 requires solving linear systems with a few basic operations, D_2 requires solving simple equations, and D_1 has one or two basic operations with clearly written numbers. We show an example from each group in the supplementary material. We didn't observe a similar pattern for BERT with the exception of the easiest group D_1 where the model chooses the answer that is somewhat different from other candidates. We provide an example of each group in the supplementary materials.

Finally, we also compare the difficulty of questions with the difficulty perceived by humans. For that, we have conducted a small-group human study, where we have asked participants to solve some AQuA-RAT questions and rate their difficulty. We find a positive correlation between the difficulty measured by our models (as described above) to the difficulty judged by humans. We give more details in the appendix.

Conclusions. We have investigated if BERT (Devlin et al., 2019) – a pre-trained, large language model – can deal with mathematical reasoning. We find that its representation is biased (Brown et al., 2020; Bhardwaj et al., 2020; Kurita et al., 2019) also in mathematics. We investigate and describe that bias. Our novel pretext tasks and losses reduce that bias, but the network still finds shortcuts. We hope our work will spark interest of the community in developing language models capable of mathematical reasoning.

Acknowledgements. We thank Wang Ling (DeepMind) for his comments and suggestions on our draft. Also, we thank Piotr Biliński and all participants of the 2020 Machine Learning Project course at the University of Warsaw for the conversations about the project. All experiments were performed using the Entropy cluster funded by NVIDIA, Intel, the Polish National Science Center grant UMO-2017/26/E/ST6/00622 and ERC Starting Grant TOTAL. The work of Henryk Michalewski was supported by the Polish National Science Center grant UMO-2018/29/B/ST6/02959.

Impact Statement

Our research follows the data-driven paradigm for creating general-purpose language models with some mathematical skills. We expect that mathematically aware language models will broaden the spectrum of topics they can understand, increasing their reliability and making them more useful.

Improving mathematical abilities and coherence in language models is likely to affect question-answering or dialogue systems, search engines or text summarization systems.

One considerable risk in developing language models at scale is that they could use various workarounds and biases to achieve their results. We have shown that issues in the context of mathematical reasoning. Such problems can become hazardous when wrong numbers could lead to bad decisions. Additionally, a person could easily fall into the fallacy that the order of magnitude is correct even if the answer is incorrect. As we showed, the model can favour round numbers over the ones close to the right answer. To mitigate the risk, we encourage considering additional tests and investigating the models more rigorously.

References

Stephan Alaniz and Zeynep Akata. 2019. Explainable observer-classifier for explainable binary decisions. *arXiv preprint arXiv:1902.01780*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Rishabh Bhardwaj, Navonil Majumder, and Soujanya Poria. 2020. Investigating gender bias in bert. *arXiv preprint arXiv:2009.05021*.

Daniel G Bobrow. 1964. Natural language input for a computer problem solving system.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*.

Eugene Charniak. 1969. Computer solution of calculus word problems. In *Proceedings of the 1st international joint conference on Artificial intelligence*, pages 303–316.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Rodney R Cocking, Rodney T Cocking, and Jose P Mestre. 1988. *Linguistic and cultural influences on learning mathematics*. Psychology Press.

Mark G Core, H Chad Lane, Michael Van Lent, Dave Gomboc, Steve Solomon, and Milton Rosenberg. 2006. Building explainable artificial intelligence systems.

Scott Crossley, Tiffany Barnes, Collin Lynch, and Danielle S McNamara. 2017. Linking language to math success in an on-line course. *International Educational Data Mining Society*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hubert L Dreyfus, L Hubert, et al. 1992. *What computers still can't do: A critique of artificial reason*. MIT press.

John Rupert Firth. 1961. *Papers in Linguistics 1934-1951: Repr*. Oxford University Press.

Lynn S Fuchs, Douglas Fuchs, Donald L Compton, Sarah R Powell, Pamela M Seethaler, Andrea M Capizzi, Christopher Schatschneider, and Jack M Fletcher. 2006. The cognitive correlates of third-grade skill in arithmetic, algorithmic computation, and arithmetic word problems. *Journal of Educational Psychology*, 98(1):29.

Lynn S Fuchs, Douglas Fuchs, Karla Stuebing, Jack M Fletcher, Carol L Hamlett, and Warren Lambert. 2008. Problem solving and computational skill: Are they shared or distinct aspects of mathematical cognition? *Journal of educational psychology*, 100(1):30.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Caglar Gulcehre, Sarath Chandar, Kyunghyun Cho, and Yoshua Bengio. 2016. Dynamic neural turing machine with soft and hard addressing schemes. *arXiv preprint arXiv:1607.00036*.

Hossein Hajipour, Mateusz Malinowski, and Mario Fritz. 2020. Ireen: Iterative reverse-engineering of black-box functions via neural program synthesis. *arXiv preprint arXiv:2006.10720*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision (ECCV)*, pages 630–645. Springer.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer.

Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. 2018. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 771–787.

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896, Berlin, Germany. Association for Computational Linguistics.

W Lewis Johnson. Agents that learn to explain themselves.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. *arXiv preprint arXiv:1906.07337*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

H Chad Lane, Mark G Core, Michael Van Lent, Steve Solomon, and Dave Gomboc. 2005. Explainable artificial intelligence for training and tutoring. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY CA INST FOR CREATIVE ....

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Percy Liang. 2016. Learning executable semantic parsers for natural language understanding. *Communications of the ACM*, 59(9):68–76.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Meghann Lomas, Robert Chevalier, Ernest Vincent Cross, Robert Christopher Garrett, John Hoare, and Michael Kopack. 2012. Explaining robot actions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 187–188.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. 2018. Learning visual question answering by bootstrapping hard attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, pages 1682–1690.

Jose P Mestre. 2013. The role of language comprehension in mathematics and problem solving. In *Linguistic and cultural influences on learning mathematics*, pages 201–220. Taylor and Francis.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.

Richard J Murnane, John B Willett, M Jay Braatz, and Yves Duhaldeborde. 2001. Do different dimensions of male high school students' skills predict labor market success a decade later? evidence from the nlsy. *Economics of Education Review*, 20(4):311–320.

Emilio Parisotto, Abdel-rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. 2016. Neuro-symbolic program synthesis. *arXiv preprint arXiv:1611.01855*.

Maria Chiara Pasolunghi, Cesare Cornoldi, and Stephanie De Liberto. 1999. Working memory and intrusions of irrelevant information in a group of specific poor problem solvers. *Memory & Cognition*, 27(5):779–790.

Markus N Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. 2020. Mathematical reasoning via self-supervised skip-tree training. *arXiv preprint arXiv:2006.04757*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1142.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Amber Y Wang, Lynn S Fuchs, and Douglas Fuchs. 2016. Cognitive and linguistic predictors of mathematical word problems with and without irrelevant information. *Learning and individual differences*, 52:79–87.

Lei Wang, Dongxiang Zhang, Lianli Gao, Jingkuan Song, Long Guo, and Heng Tao Shen. 2018. Mathdqn: Solving arithmetic word problems via deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Yanyan Zou and Wei Lu. 2019. Text2Math: End-to-end parsing text into math expressions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5327–5337, Hong Kong, China. Association for Computational Linguistics.

A AQuA-RAT example

Question: *A starts a business with Rs.40,000. After 2 months, B joined him with Rs.60,000. C joined them after some more time with Rs.120,000. At the end of the year, out of a total profit of Rs.375,000, C gets Rs.150,000 as his share. How many months after B joined the business, did C join?*
Options: *A) 30, B) 32, C) 35, D) 36, E) 40*
Rationale:

Assume that C was there in the business for x months

$$A : B : C = 40000 * 12 : 60000 * 10 : 120000 * x$$
$$= 40 * 12 : 60 * 10 : 120x = 40 : 5 * 10 : 10x$$
$$= 8 : 10 : 2x$$
$$= 4 : 5 : x$$
$$\textit{C's share} = 375000 * x/(9 + x) = 150000$$
$$=> 375x/(9 + x) = 150$$
$$=> 15x = 6(9 + x)$$
$$=> 5x = 18 + 2x$$
$$=> 3x = 18$$
$$=> x = 18/3 = 6$$

It means C was there in the business for 6 months. Given that B joined the business after 2 months. Hence C joined after 4 months after B joined
Answer is B

Additional examples are in the supplementary material.

B Input representation

All BERT variants use the representation that corresponds to a special token [CLS] that we put at the beginning of the whole input sequence consisting of question tokens followed by rationale tokens, and in the downstream, question-answering task, rationale tokens are replaced by the answer options. With that, the classification uses the contextual embedding of [CLS] that captures the entire input. MLM classifies over the entire vocabulary of possible words while the other two losses use a binary cross-entropy loss for the predictions.

C Training protocol

We train all our architectures on AQuA-RAT using the following training phases. In all cases, we choose our best model based on the performance on the validation set (dev set), and report the final performance on the test set.

Pre-training. Each model is pre-trained on a large corpus of texts written in natural language sampled from English Wikipedia and BooksCorpus (Devlin et al., 2018; Zhu et al., 2015). We use this as the base (BERT-base) model that is also used in all other variants of BERT. In practice, we initialize all the models with the weights using the HuggingFace library (Wolf et al., 2019) and don't keep final layer for fine-tuning. Our model therefore has the same number of weights as BERT-base.

Self-supervision. Here, we use our newly introduced losses, ROP and NROP, where our models use questions and possibly rationales from the AQuA-RAT dataset. Both questions and rationales use the same word embeddings. However, to distinguish between both modalities we use two segment embeddings. The first one for all the question tokens, and the second one for all the rationale tokens. That is, the segment embedding is shared among all the question tokens, and separately among all the rationale tokens. We use dynamic masking (Liu et al., 2019). Here, tokens are randomly masked for each batch. We naturally extend this approach to other losses that we use in this phase. That is, ROP and NROP negative examples are randomly recreated every k epochs, where $k = 2$ in our case.

Fine-tuning is the last training phase. Here, once our models have learnt the representation during the self-supervised phase, we tune such a representation to the question-answering downstream task. In this task, our input consists of question tokens and possible answer options. There are five such options that comes with the dataset. Like other methods, we tread this as a five-class classification task where the classification head is added on top of the final embedding of the input. We consider the embedding corresponding to the first (from the left) [CLS] token as such the final representation.

D Implementation details

In our experiments, we use four TITAN V GPUs. We use a multi-gpu setup. In the pre-training phase, we use batch size equals to four for each GPU device. Therefore the effective batch size equals to sixteen. We use the learning rate $5 \cdot 10^{-5}$ and trained the models for 24 epochs. In the fine-tuning phase, we use early stopping criteria, based on the accuracy score on the validation set. We use the following criteria. If the model does not improve the performance in 15 consecutive epochs, we stop training, and evaluate a model that yields the highest validation performance. We use ADAM optimizer with learning rate 10^{-5} and gradient clipping that sets the maximal gradient's norm to one. All our settings use the same hyper-parameters but they differ due to the random initialization of our self-supervised networks (during the self-supervised training phase) and the classification networks (during the fine-tuning phase). Self-supervision phase takes around 4 days on 4 GPUs, whereas fine-tuning takes 8 hours on a single GPU.

E Permutation invariant methods

In the main paper, we have shown that typical models can use positional biases in achieving answers. This results in a low permutation consistency score (Table 3 in the main paper). To handle that issue, we have defined extra variants that do not use positional encodings for the answer options and instead they rely on the retrieval mechanics where input representations are matched against the candidate answers. Here, we describe two such variants.

E.1 Original methods

Original models create an embedding of a sentence extended by possible questions. This embedding is then transformed by a linear layer to predict the correct answer. That is,

$$o_1 = f_1(\textbf{BERT}(Q||P))$$

where o_1 is a 5-dimensional vector with probabilities for each possible answer, Q is a question, P are all possible answers, $||$ represents concatenation, f_1 is a single fully connected layer from 768-dimensional space to 5-dimensional space with the softmax activation. **BERT** is a BERT-base sentence embedding. The same approach is used for BERT+(N)ROP.

E.2 SEP-NC

In SEP-NC and SEP-C, we use separate embeddings for a question and **SEP**arate embedding for a candidate answer. They differ, however, in the fact that SEP-C has access to all five possible answers, while SEP-NC has access only to one prompted candidate answer. Therefore NC stands for ńo candidates", while C stands for ćandidates".

We train the SEP-NC model on a binary classification task to predict whether each candidate answer C is correct. The method produces two embeddings, one for question and another one for a candidate answer $C \in P$, and next concatenates them. That is,

$$o_2 = f_2(\textbf{BERT}(Q)||\textbf{BERT}(C))$$

where o_2 is an estimated probability that C is a correct answer, P is the sequence of all possible answers, f_2 is a single fully connected layer from 1536 (768 * 2) dimensional space to 1-dimensional space with the sigmoid activation. Note that, all candidate answers are independent of the question. That is, BERT cannot use positional biases in deriving an answer. At test time, the model is prompted to score all five candidate answers and select the one with the highest score. We naturally extended that approach to BERT+ROP and BERT+NROP. Table 3 (the main paper) shows a significant improvement over the baseline method.

E.3 SEP-C

SEP-NC method could be too restrictive as it does not allow the model to compare against different answers. Therefore, we propose another approach that 1) alleviate the issue with positional biases, but 2) can compare between different answer options. We call that approach SEP-C.

Originally for each token, a positional encoding is assigned based on its position. In SEP-C, before assigning positional encoding, we artificially reset the position at the beginning of each possible answer. For example, if possible answers are: $a)10, b)20, c)30, d)40, e)50$ they are changed into $10; 20; 30; 40; 50$ and after the tokenization, we get the following list of tokens: ['1','0', ';', '2', '0', ';', '3', '0', ';' ,'4', '0', ';', '5', '0']. Modified positional encoding will assign value based only on the relative position to the beginning of the current possible answer. Therefore, in the example above, each '0' will receive the same positional encoding, and '1' will get the same positional encoding as '2', '3', and so on.

Formally, we have

$$o_3 = f_3(\textbf{BERT}(Q||P_m)||\textbf{BERT}(C))$$

where P_m is the sequence of all the possible answers but modified as explained above. Note that, in this formulation, the model can use the information for all the possible answer options, but their order is not taken into account. Table 3 (the main paper) shows a significant improvement over the baseline method.

E.4 Human study

We carried an initial human study on the group of 16 volunteers from University of Warsaw. Volunteers were Mathematics and Informatics students from the Faculty of Mathematics, Informatics and Mechanics. We asked the participants to solve questions sampled from the AQuA-RAT dataset. We are interested in the relation between BERTs difficulty, BERT+NROP difficulty and human difficulty. Therefore to have a full image we would like to have 2 questions for each question difficulty pair, for example (D_1 BERT, D_2**:** BERT+NROP)

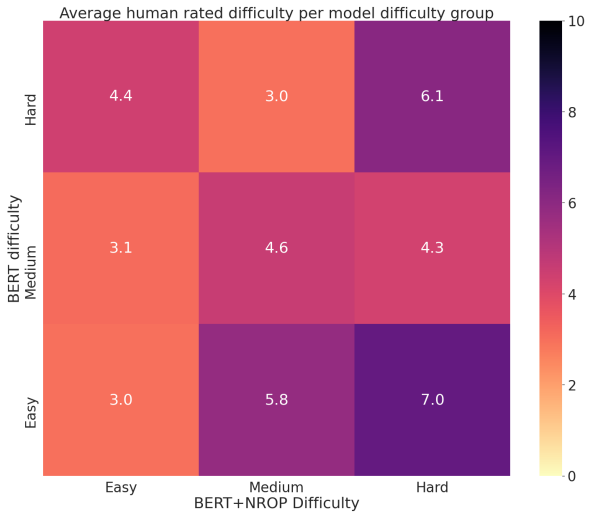


Figure 6: The average human-judged difficulty for questions from each model difficulty group.

. However, that would give 25 combinations and 50 questions if we wanted to have 2 questions per combination. That would be too much to ask from a volunteer participant. In order to reduce the number of questions, we group our 5 difficulty groups into 3 categories as follows.

- Easy: D_1
- Medium: D_2 and D_3 combined
- Hard: D_4 and D_5 combined

Because of that we have only 9 possible combinations and by sampling 2 questions from each combination we still have a feasible number of questions (18).

Apart from solving the question, we asked to rate question difficulty on a scale from 1 (the simplest) to 10 (the most challenging). In general, our participants were knowledgeable in math and solved all the questions correctly. With that grouping we now

The average human-rated difficulty for each of 9 combinations is presented in Figure 6. The results show that the progression of human difficulty is correlated with the difficulty judged by the models. Additionally, the human difficulty seems to be more sensitive to BERT+NROP difficulty than to BERTs. In other words, increasing the difficulty of BERT+NROP will increase the human difficulty more than the increasing difficulty of BERT. This observation fits our previous observations that BERT+NROP solves the most straightforward questions while BERT is looking for some leaks, like looking for the roundest answer.

dataset	A	B	C	D	E
train	21.03%	22%	22.87%	19.95%	14.15%
dev	27.17%	25.98%	16.93%	19.69%	10.24$
test	24.80%	22.83%	20.87%	18.11%	13.38%

Table 4: Answer distribution in each dataset.

F Distribution of answers

Table 4 shows the distribution of the answers in the AQuA-RAT (Ling et al., 2017) dataset in all the folds. Imbalance in distributions could potentially be used by models to find easy, shortcut solutions. For instance, a constant classifier that always choose the first answer (A) gets about 24% test accuracy.

G Negative results

While developing our self-supervised losses, we have developed another loss that turned out to be unhelpful. Here, we describe that loss as some its parts could be insightful for others. (N)ROP is a local loss focusing on rationales but not on the connections between questions and rationales. For that, we have developed Question Rationale Alignment (QRA). QRA changes a rationale with 50% probability to a randomly chosen rationale from the current batch. However, simply changing rationales would result in trivially solvable task in most cases. All the model would have to do is check whether numbers in the rationale and the question match. Hence, we mask number tokens with a special token QRA alone or QRA combined with NROP does not improve the results, it gives it gives 33.9% accuracy on the test in the best combination, so we didn't include it in the main results.

H Related work

We are inspired by the following research.

BERTology. We use BERT (Devlin et al., 2019) as our core. It uses Transformers (Vaswani et al., 2017); powerful neural architectures that applies a trainable function to all the pairs of input embeddings. It also uses masking that covers a fraction of the input words and requires the network to predict the hidden words based on the context. With both ingredients, the meaning (representation) of a word emerges from the "company it keeps" (Firth, 1961). In practice, often, such representations are pre-trained on large textual corpora with no need for annotations, and next fine-tuned on the downstream tasks. BERT's strong performance has resulted in the Cambrian explosion of studies of the

inner working mechanisms and various modifications (Clark et al., 2019; de Vries et al., 2019; Lan et al., 2019; Liu et al., 2019; Sanh et al., 2019; Radford et al.; Raffel et al., 2019; Yang et al., 2019). Finally, our Reasoning Order Prediction (ROP) is inspired by Sentence Order Prediction (SOP) (Lan et al., 2019). However, ROP works with multiple rationale sentences, where by changing the order we force the network to understand the consecutive "reasoning" steps. We have also further extended ROP to a more difficult Neighbor Reasoning Order Prediction (NROP).

Language and math. Development psychologists (Cocking et al., 1988; Mestre, 2013) often argue for the necessity of learning languages and point out that those with limited language skills are in danger of under-performing at school. Moreover, it is also believed that language studies involve discipline in learning and manipulating formal structures, and thus may promote the development of the organization of thoughts also required in mathematical reasoning. The similarity between linguistic competence and mathematics is especially pronounced when solving math word problems (Fuchs et al., 2006, 2008; Wang et al., 2016). Interestingly, attention appears to be crucial in problem solving (Fuchs et al., 2006; Pasolunghi et al., 1999). (Crossley et al., 2017) show that language skills are correlated with the performance in mathematical tests also among the university students. In particular, they pointed out that ability to use complex syntactic structures and cohesion devices are linked to better scores in a blended discrete mathematics course. We take inspiration from all such studies and decide to build our mathematical model based on language models.

Math word problems. Solving math word problems is a significant component of the mathematics curriculum and is taught very early, thoroughly, and universally. Such the emphasize is often motivated by that solving them is among the best predictors of employability, and is considered as a distinct area of mathematical competence (Murnane et al., 2001; Wang et al., 2016). Since solving such problems is unique to human intelligence, math word problems are also interesting for the AI community. This results in various approaches, more traditional symbolic methods, neural networks, and neuro-symbolic methods. (Bobrow, 1964; Charniak, 1969; Shi et al., 2015; Ling et al., 2017; Amini et al., 2019; Parisotto et al., 2016; Wang et al., 2018; Zou and Lu, 2019) as well as datasets (Ling et al., 2017; Amini et al., 2019; Huang et al., 2016; Saxton et al., 2019) An interesting approach is proposed in (Rabe et al., 2020), in which authors use self-supervised tasks on parsing trees of formal expressions. This approach requires syntax trees, and hence we would have to use an external parser. As our goal was to make an end to end model, we did not experiment with it, but there are no obstacles against using it in symbiosis with our methods. (Geva et al., 2020) also proposes self-supervised training for improving mathematical abilities in language models. We, however, focused on a data-driven approach to exclude choice biases and therefore restricted ourselves from using generated data.

Rationales. In human communication, we always expect there is some rationale behind each decision. Hence, we set the same expectations to our artificial agents. Symbolic or semi-symbolic architectures naturally produce justifications as a sequence of formulas in some formal language (Lane et al., 2005; Core et al., 2006; Lomas et al., 2012; Johnson; Liang, 2016; Malinowski and Fritz, 2014). Ideally, such rationales would also be shared and communicated to us through some language. The latter approach is especially appealing when applied to black-box neural networks. For instance, (Hendricks et al., 2016) propose a system that classifies the input image as well as it produces a textual explanation on "why this class is suitable for the given image".

Systems that produce explanations either in the form of the language (Ling et al., 2017; Hendricks et al., 2016), attention (Bahdanau et al., 2014; Mnih et al., 2014; Gulcehre et al., 2016; Malinowski et al., 2018; Xu and Saenko, 2016; Yang et al., 2016), phrase selection (Lei et al., 2016), distillation into programs (Hajipour et al., 2020), or decision trees (Alaniz and Akata, 2019) can potentially increase the transparency of the black-box neural networks. However, most of these approaches create rationales posthoc where the justification is conditioned on answers or by querying the network. In our work, we use rationales to learn a finer representation that can potentially lead to better decisions. In this sense, our technique is conceptually closer to methods that derive answers based on the program and use rationales paired with questions to guide the program induction process (Ling et al., 2017).

Happy Dance, Slow Clap: Using Reaction GIFs to Predict Induced Affect on Twitter

Boaz Shmueli[1,2,3] , **Soumya Ray**[2], and **Lun-Wei Ku**[3]
[1]Social Networks and Human-Centered Computing, TIGP, Academia Sinica
[2]Institute of Service Science, National Tsing Hua University
[3]Institute of Information Science, Academia Sinica
shmueli@iis.sinica.edu.tw soumya.ray@iss.nthu.edu.tw lwku@iis.sinica.edu.tw

Abstract

Datasets with *induced emotion* labels are scarce but of utmost importance for many NLP tasks. We present a new, automated method for collecting texts along with their *induced reaction* labels. The method exploits the online use of reaction GIFs, which capture complex affective states. We show how to augment the data with *induced emotions* and *induced sentiment* labels. We use our method to create and publish ReactionGIF, a first-of-its-kind affective dataset of 30K tweets. We provide baselines for three new tasks, including induced sentiment prediction and multilabel classification of induced emotions. Our method and dataset open new research opportunities in emotion detection and affective computing.

1 Introduction

Affective states such as emotions are an elemental part of the human condition. The automatic detection of these states is thus an important task in affective computing, with applications in diverse fields including psychology, political science, and marketing (Seyeditabari et al., 2018). Training machine learning algorithms for such applications requires large yet task-specific emotion-labeled datasets (Bostan and Klinger, 2018).

Borrowing from music (Gabrielsson, 2001) and film (Tian et al., 2017), one can distinguish between two reader perspectives when labeling emotions in text: *perceived* emotions, which are the emotions that the reader recognizes in the text, and *induced* emotions, which are the emotions aroused in the reader. However, with the exception of Buechel and Hahn (2017), this distinction is mostly missing from the NLP literature, which focuses on the distinction between author and reader perspectives (Calvo and Mac Kim, 2013).

The collection of perceived emotions data is considerably simpler than induced emotions data, and presently most human-annotated emotion datasets are labeled with perceived emotions (e. g., Strapparava and Mihalcea, 2008; Preoţiuc-Pietro et al., 2016; Hsu and Ku, 2018; Demszky et al., 2020). Induced emotions data can be collected using physiological measurements or self-reporting, but both methods are complex, expensive, unreliable and cannot scale easily. Still, having well-classified induced emotions data is of utmost importance to dialogue systems and other applications that aim to detect, predict, or elicit a particular emotional response in users. Pool and Nissim (2016) used distant supervision to detect induced emotions from Facebook posts by looking at the six available emoji reactions. Although this method is automatic, it is limited both in emotional range, since the set of reactions is small and rigid, and accuracy, because emojis are often misunderstood due to their visual ambiguity (Tigwell and Flatla, 2016).

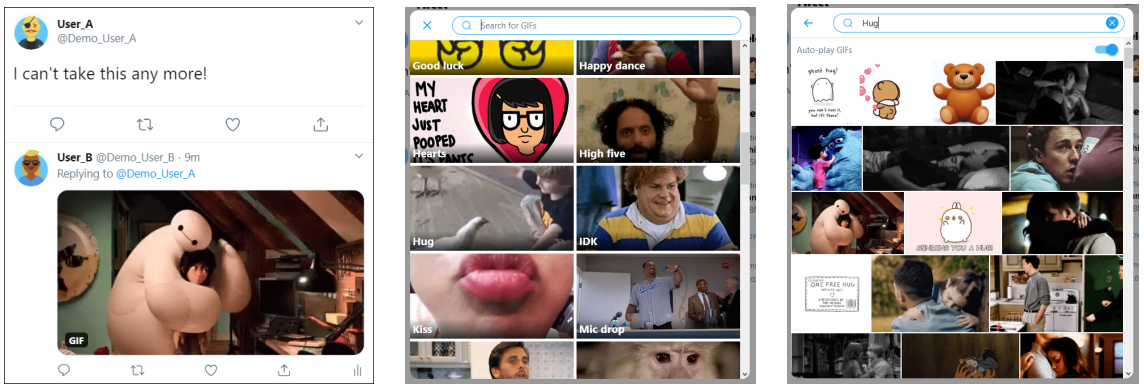
To overcome these drawbacks, we propose a new method that innovatively exploits the use of reaction GIFs in online conversations. Reaction GIFs are effective because they "display emotional responses to prior talk in text-mediated conversations" (Tolins and Samermit, 2016). We propose a fully-automated method that captures in-the-wild texts, naturally supervised using *fine-grained*, *induced reaction* labels. We also augment our dataset with sentiment and emotion labels. We use our method to collect and publish the ReactionGIF dataset.[1]

2 Automatic Supervision using GIFs

Figure 1a shows a typical Twitter thread. User *A* writes *"I can't take this any more!"*. User *B* replies with a reaction GIF depicting an embrace. Our method automatically infers a *hug* reaction, signaling that *A*'s text induced a feeling of love and caring. In the following, we formalize our method.

[1] github.com/bshmueli/ReactionGIF

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 395–401
August 1–6, 2021. ©2021 Association for Computational Linguistics



(a) A root tweet ("I can't take...") with a *hug* reaction GIF reply.

(b) The reaction categories menu offers 43 categories (*hug*, *kiss*, ...).

(c) The top reaction GIFs offered to the user from the *hug* category.

Figure 1: How reaction GIFs are used (left) and inserted (middle, right) on Twitter.

2.1 The Method

Let (t, g) represent a 2-turn online interaction with a root post comprised solely of text t, and a reply containing only reaction GIF g. Let $R = \{R_1, R_2, ..., R_M\}$ be a set of M different *reaction categories* representing various affective states (e. g., *hug*, *facepalm*). The function $\mathfrak{R}$ maps a GIF g to a reaction category, $g \mapsto \mathfrak{R}(g)$, $\mathfrak{R}(g) \in R$. We use $r = \mathfrak{R}(g)$ as the label of t. In the Twitter thread shown in Figure 1a, the label of the tweet *"I can't take this any more!"* is $r = \mathfrak{R}(g) =$ *hug*.

Inferring $\mathfrak{R}(g)$ would usually require humans to manually view and annotate each GIF. Our method automatically determines the reaction category conveyed in the GIF. In the following, we explain how we automate this step.

GIF Dictionary We first build a dictionary of GIFs and their reaction categories by taking advantage of the 2-step process by which users post reaction GIFs. We describe this process on Twitter; other platforms follow a similar approach:

Step 1: The user clicks on the GIF button. A menu of reaction categories pops up (Figure 1b). Twitter has 43 pre-defined categories (e. g., *high five*, *hug*). The user clicks their preferred category.

Step 2: A grid of reaction GIFs from the selected category is displayed (Figure 1c). The user selects one reaction GIF to insert into the tweet.

To compile the GIF dictionary, we collect the first 100 GIFs in each of the $M = 43$ reaction categories on Twitter. We save the 4300 GIFs, along with their categories, to the dictionary. While in general GIFs do not necessarily contain affective information, our method collects *reaction* GIFs that depict corresponding affective states.

Computing $\mathfrak{R}(g)$ Given a (t, g) sample, we label text t with reaction category r by mapping reaction GIF g back to its category $r = \mathfrak{R}(g)$. We search for g in the GIF dictionary and identify the category(ies) in which it is offered to the user. If the GIF is not found, the sample is discarded. For the small minority of GIFs that appear in two or more categories, we look at the positions of the GIF in each of its categories and select the category with the higher position.

2.2 Category Clustering

Because reaction categories represent overlapping affective states, a GIF may appear in multiple categories. For example, a GIF that appears in the *thumbs up* category may also appear in the *ok* category, since both express approval. Out of the 4300 GIFs, 408 appear in two or more categories. Exploiting this artefact, we propose a new metric: the pairwise *reaction similarity*, which is the number of reaction GIFs that appear in a pair of categories.

To automatically discover affinities between reaction categories, we use our similarity metric and perform hierarchical clustering with average linkage. The resulting dendrogram, shown in Figure 2, uncovers surprisingly well the relationships between common human gesticulations. For example, *shrug* and *idk* (**I d**on't **k**now) share common emotions related to uncertainty and defensiveness. In particular, we can see two major clusters capturing negative sentiment (left cluster: *mic drop* to *smh* [**s**hake **m**y **h**ead]) and positive sentiment (right cluster: *hug* to *slow clap*), which are useful for downstream sentiment analysis tasks. The two rightmost singletons, *popcorn* and *thank you*, lack sufficient similarity data.

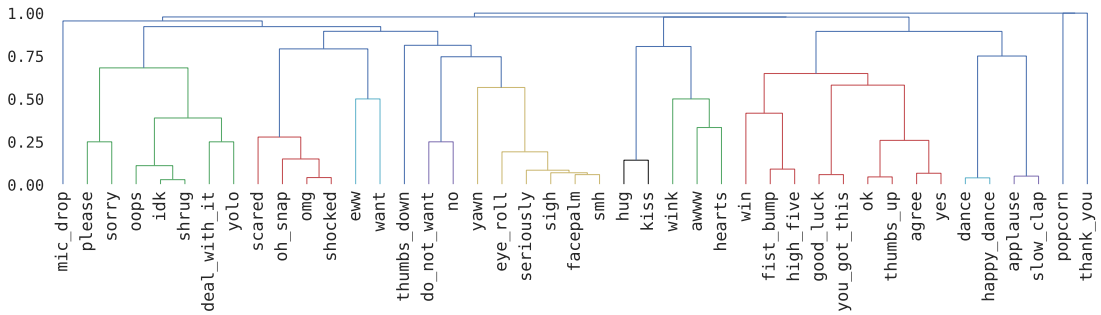


Figure 2: Hierarchical clustering (average linkage) of reaction categories shows relationships between reactions.

3 ReactionGIF Dataset

We applied our proposed method to 30K English-language (t, g) 2-turn pairs collected from Twitter in April 2020. t are text-only root tweets (not containing links or media) and g are pure GIF reactions. We label each tweet t with its reaction category $r = \mathfrak{R}(g)$. See Appendix A for samples. The resulting dataset, ReactionGIF, is publicly available.

Figure 3 shows the category distribution's long tail. The top seven categories (*applause* to *eye-roll*) label more than half of the samples (50.9%). Each of the remaining 36 categories accounts for between 0.2% to 2.8% of the samples.

Label Augmentation Reaction categories convey a rich affective signal. We can thus augment the dataset with other affective labels. We add **sentiment labels** by using the positive and negative reaction category clusters, labeling each sample according to its cluster's sentiment (§2.2). Furthermore, we add **emotion labels** using a novel reactions-to-emotions mapping: we asked 3 annotators to map each reaction category onto a subset of the 27 emotions in Demszky et al. (2020) — see Table 1. Instructions were to view the GIFs in each category and select the expressed emotions. Pairwise Cohen's kappa indicate moderate interrater agreements with $\kappa_{12} = 0.512$, $\kappa_{13} = 0.494$, $\kappa_{23} = 0.449$, and Fleiss' kappa $\kappa_F = 0.483$. We use the annotators' majority decisions as the final many-to-many mapping and label each sample according to its category's mapped emotions subset.

GIFs in Context As far as we know, our dataset is the first to offer reaction GIFs with their eliciting texts. Moreover, the reaction GIFs are labeled with a reaction category. Other available GIF datasets (TGIF by Li et al., 2016, and GIFGIF/GIFGIF+, e. g., Jou et al., 2014) lack both the eliciting texts and the reaction categories.

Admiration	Curiosity	Fear	Pride
Amusement	Desire	Gratitude	Realization
Anger	Disappointment	Grief	Relief
Annoyance	Disapproval	Joy	Remorse
Approval	Disgust	Love	Sadness
Caring	Embarrassment	Nervousness	Surprise
Confusion	Excitement	Optimism	

Table 1: The 27 emotions in Demszky et al. (2020).

4 Baselines

As this is the first dataset of its kind, we aim to promote future research by offering baselines for predicting the reaction, sentiment, and emotion induced by tweets. We use the following four models in our experiments:

- **Majority**: A simple majority class classifier.
- **LR**: Logistic regression classifier (L-BFGS solver with $C = 3$, maximum iterations 1000, stratified K-fold cross validation with $K = 5$) using TF-IDF vectors (unigrams and bigrams, cutoff 2, maximum 1000 features, removing English-language stop words).
- **CNN**: Convolutional neural network (100 filters, kernel size 3, global max pooling; 2 hidden layers with 0.2 dropout; Adam solver, 100 epochs, batch size 128, learning rate 0.0005) with GloVe embeddings (Twitter, 27B tokens, 1.2M vocabulary, uncased, 100d) (Pennington et al., 2014).
- **RoBERTa**: Pre-trained transformer model (base, batch size 32, maximum sequence length 96, 3 training epochs) (Liu et al., 2019).

We hold out 10% of the samples for evaluation. The code is publicly available along with the dataset for reproducibility. The experiment results are summarized in Table 2.

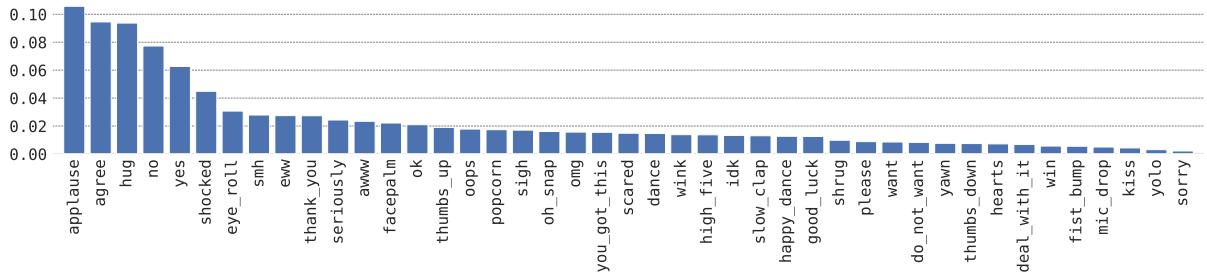


Figure 3: Distribution of the 43 reaction categories in ReactionGIF

Task →	Reaction				Sentiment				Emotion
Model ↓	Acc	P	R	F_1	Acc	P	R	F_1	LRAP
Majority	10.4	1.1	10.4	2.0	58.0	33.7	58.0	42.6	0.445
LR	22.7	19.5	22.7	18.0	64.7	64.4	64.7	62.4	0.529
CNN	25.5	17.3	25.5	19.1	67.1	66.8	67.1	66.3	0.557
RoBERTa	**28.4**	**23.6**	**28.4**	**23.9**	**70.0**	**69.7**	**70.0**	**69.8**	**0.596**

Table 2: Baselines for the reaction, sentiment, and emotion classification tasks. All metrics are weight-averaged. The highest value in each column is emboldened.

Affective Reaction Prediction is a multiclass classification task where we predict the reaction category r for each tweet t. RoBERTa achieves a weight-averaged F_1-score of 23.9%.

Induced Sentiment Prediction is a binary classification task to predict the sentiment induced by tweet t by using the augmented labels. RoBERTa has the best performance with accuracy 70.0% and F_1-score of 69.8%.

Finally, **Induced Emotion Prediction** uses our reaction-to-emotion transformation for predicting emotions. This is a 27-emotion *multilabel* classification task, reflecting our dataset's unique ability to capture complex emotional states. RoBERTa is again the best model, with Label Ranking Average Precision (LRAP) of 0.596.

5 Discussion

Reaction GIFs are ubiquitous in online conversations due to their uniqueness as lightweight and silent moving pictures. They are also more effective and precise[2] when conveying affective states compared to text, emoticons, and emojis (Bakhshi et al., 2016). Consequently, the reaction category is a new type of label, not yet available in NLP emotion datasets: existing datasets use either the discrete emotions model (Ekman, 1992) or the dimensional model of emotion (Mehrabian, 1996). The new labels possess important advantages, but also present interesting challenges.

Advantages The new reaction labels provide a rich, complex signal that can be mapped to other types of affective labels, including sentiment, emotions and possibly feelings and moods. In addition, because reaction GIFs are ubiquitous in online conversations, we can automatically collect large amounts of inexpensive, naturally-occurring, high-quality affective labels. Significantly, and in contrast with most other emotion datasets, the labels measure *induced* (as opposed to *perceived*) affective states; these labels are of prime importance yet the most difficult to obtain, with applications that include GIF recommender systems, dialogue systems, and any other application that requires predicting or inducing users' emotional response.

Challenges The large number of reaction categories (reflecting the richness of communication by gestures) makes their prediction a challenging task. In addition, the category distribution has a long tail, and there is an affective overlap between the categories. One way to address these issues is by accurately mapping the reactions to emotions. Precise mapping will require a larger GIF dictionary (our current one has 4300 GIFs), a larger dataset, and new evaluation metrics. A larger GIF dictionary will also improve the *reaction similarity*'s accuracy, offering new approaches for studying relationships between reactions (§2.2).

[2]For example, the *facepalm* reaction is "a gesture in which the palm of one's hand is brought to one's face, as an expression of disbelief, shame, or exasperation.", Oxford University Press, lexico.com/en/definition/facepalm

6 Conclusion

Our new method is the first to exploit the use of reaction GIFs for capturing in-the-wild *induced* affective data. We augment the data with induced sentiment and emotion labels using two novel mapping techniques: reaction category clustering and reactions-to-emotions transformation. We used our method to publish ReactionGIF, a first-of-its-kind dataset with multiple affective labels. The new method and dataset offer opportunities for advances in emotion detection.

Moreover, our method can be generalized to capture data from other social media and instant messaging platforms that use reaction GIFs, as well as applied to other downstream tasks such as multimodal emotion detection and emotion recognition in dialogues, thus enabling new research directions in affective computing.

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. Special thanks to Thilina Rajapakse, creator of the elegant Simple Transformers package, for his help.

This research was partially supported by the Ministry of Science and Technology in Taiwan under grants MOST 108-2221-E-001-012-MY3 and MOST 109-2221-E-001-015- [*sic*].

Ethical Considerations and Implications

Data Collection

The ReactionGIF data was collected from Twitter using the official API in full accordance with their Development Agreement and Policy (Twitter, 2020). Similar to other Twitter datasets, we include the tweet IDs but not the texts. This guarantees that researchers who want to use the data will also need to agree with Twitter's Terms of Service. It also ensures compliance with section III (Updates and Removals) of the Developer Agreement and Policy's requirement that when users delete tweets (or make them private), these changes are reflected in the dataset (Belli et al., 2020).

Annotation

Annotation work was performed by three adult students, two males and one female, who use social media regularly. The labeling involved viewing 43 sets of standard reaction GIFs, one for each reaction category. These reaction GIFs are the standard offering by the Twitter platform to all its users. As a result, this content is highly familiar to users of social media platforms such as Facebook or Twitter, and thus presents a very low risk of psychological harm. Annotators gave informed consent after being presented with details about the purpose of the study, the procedure, risks, benefits, statement of confidentiality and other standard consent items. Each annotator was paid US$18. The average completion time was 45 minutes.

Applications

The dataset and resulting models can be used to infer readers' induced emotions. Such capability can be used to help online platforms detect and filter out content that can be emotionally harmful, or emphasize and highlight texts that induce positive emotions with the potential to improve users' well-being. For example, when a person is in grief or distress, platforms can give preference to responses which will induce a feeling of caring, gratitude, love, or optimism. Moreover, such technology can be of beneficial use in assistive computing applications. For example, people with emotional disabilities can find it difficult to understand the emotional affect in stories or other narratives, or decipher emotional responses by third parties. By computing the emotional properties of texts, such applications can provide hints or instructions and provide for smoother and richer communication. However, this technology also has substantial risks and peril. Inducing users' affective response can also be used by digital platforms in order to stir users into specific action or thoughts, from product purchase and ad clicking to propaganda and opinion forming. Deployers must ensure that users understand and agree to the use of such systems, and consider if the benefit created by such systems outweigh the potential harm that users may incur.

References

Saeideh Bakhshi, David A. Shamma, Lyndon Kennedy, Yale Song, Paloma de Juan, and Joseph 'Jofish' Kaye. 2016. Fast, cheap, and good: Why animated GIFs engage us. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 575–586, New York, NY, USA. Association for Computing Machinery.

Luca Belli, Sofia Ira Ktena, Alykhan Tejani, Alexandre Lung-Yut-Fong, Frank Portman, Xiao Zhu, Yuanpu Xie, Akshay Gupta, Michael M. Bronstein, Amra Delić, Gabriele Sottocornola, Vito Walter Anelli,

Nazareno Andrade, Jessie Smith, and Wenzhe Shi. 2020. Privacy-preserving recommender systems challenge on Twitter's home timeline. *CoRR*, abs/2004.13715.

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sven Buechel and Udo Hahn. 2017. Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 1–12, Valencia, Spain. Association for Computational Linguistics.

Rafael A. Calvo and Sunghwan Mac Kim. 2013. Emotions in text: Dimensional and categorical models. *Computational Intelligence*, 29(3):527–543.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200.

Alf Gabrielsson. 2001. Emotion perceived and emotion felt: Same or different? *Musicae Scientiae*, 5(Special Issue: Current Trends in the Study of Music and Emotion):123–147.

Chao-Chun Hsu and Lun-Wei Ku. 2018. SocialNLP 2018 EmotionX challenge overview: Recognizing emotions in dialogues. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 27–31, Melbourne, Australia. Association for Computational Linguistics.

Brendan Jou, Subhabrata Bhattacharya, and Shih-Fu Chang. 2014. Predicting viewer perceived emotions in animated GIFs. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 213–216, New York, NY, USA. Association for Computing Machinery.

Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. 2016. TGIF: A new dataset and benchmark on animated GIF description. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4641–4650, Los Alamitos, CA, USA. IEEE Computer Society.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Albert Mehrabian. 1996. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Chris Pool and Malvina Nissim. 2016. Distant supervision for emotion detection using Facebook reactions. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, pages 30–39, Osaka, Japan. The COLING 2016 Organizing Committee.

Daniel Preoţiuc-Pietro, H. Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elisabeth Shulman. 2016. Modelling valence and arousal in Facebook posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 9–15, San Diego, California. Association for Computational Linguistics.

Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. 2018. Emotion detection in text: a review. *arXiv preprint arXiv:1806.00674*.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, page 1556–1560, New York, NY, USA. Association for Computing Machinery.

Leimin Tian, Michal Muszynski, Catherine Lai, Johanna D. Moore, Theodoros Kostoulas, Patrizia Lombardo, Thierry Pun, and Guillaume Chanel. 2017. Recognizing induced emotions of movie audiences: Are induced and perceived emotions the same? In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 28–35.

Garreth W. Tigwell and David R. Flatla. 2016. Oh that's what you meant!Reducing emoji misunderstanding. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '16, page 859–866, New York, NY, USA. Association for Computing Machinery.

Jackson Tolins and Patrawat Samermit. 2016. GIFs as embodied enactments in text-mediated conversation. *Research on Language and Social Interaction*, 49(2):75–91.

Twitter. 2020. Developer Agreement and Policy. `https://developer.twitter.com/developer-terms/agreement-and-policy`. (Accessed on 02/01/2021).

Record ID	Tweet	GIF Response	Reaction Category	Sentiment	Emotions
13241	“so...I have a job now 😊”		dance	positive	Amusement, Excitement, Joy
1320	“dyed my hair..... Pics soon”		applause	positive	Admiration Approval Excitement Gratitude Surprise
17	“Don't forget to Hydrate!”		yawn	negative	Disappointment Disapproval
808	“Folks, I have a BIG BIG announcement coming tomorrow night at 9 PM EST”		scared	negative	Confusion Fear Nervousness Surprise

Figure 4: ReactionGIF samples.

A Dataset Samples

Figure 4 includes four samples from the dataset. For each sample, we show the record ID within the dataset, the text of the tweet, a thumbnail of the reaction GIF, the reaction category of the GIF, and the two augmented labels: the sentiment and the emotions.

Exploring Listwise Evidence Reasoning with T5 for Fact Verification

Kelvin Jiang, Ronak Pradeep, and **Jimmy Lin**
David R. Cheriton School of Computer Science
University of Waterloo
{kelvin.jiang, rpradeep, jimmylin}@uwaterloo.ca

Abstract

This work explores a framework for fact verification that leverages pretrained sequence-to-sequence transformer models for sentence selection and label prediction, two key sub-tasks in fact verification. Most notably, improving on previous pointwise aggregation approaches for label prediction, we take advantage of T5 using a listwise approach coupled with data augmentation. With this enhancement, we observe that our label prediction stage is more robust to noise and capable of verifying complex claims by jointly reasoning over multiple pieces of evidence. Experimental results on the FEVER task show that our system attains a FEVER score of 75.87% on the blind test set. This puts our approach atop the competitive FEVER leaderboard at the time of our work, scoring higher than the second place submission by almost two points in label accuracy and over one point in FEVER score.

1 Introduction

In recent years, the Internet has become an effective platform for creating and sharing content to large audiences. Unfortunately, there have been occurrences of bad actors taking advantage of this to propagate manipulative information for their benefit, often to the point of spreading misinformation. With the large amount of data being generated on the Internet each day, it is infeasible to manually verify it all, motivating recent research into automated fact verification.

In this work, we explore a fact verification framework built with the pretrained sequence-to-sequence transformer T5 (Raffel et al., 2020) as its backbone which we call LisT5. Within a standard three-stage architecture, we focus mostly on the label prediction problem. We adopt a "listwise approach", where all candidate sentences that form the evidence set of a claim are considered together. Our main contribution is a data augmentation technique that involves deliberately introducing noise into training data to combat data sparsity and produce a more robust model. At its introduction, a full pipeline using our techniques represents the state of the art, achieving the top scoring run on the FEVER leaderboard. An additional minor contribution exploits named entities during the sentence selection phase, which has a small but noticeable effect on generating a better candidate set for downstream label prediction. We believe that these techniques can be potentially valuable to a broader range of NLP tasks that also involve aggregation of information from upstream retrieval models.

2 Background and Related Work

As this work focuses on the Fact Extraction and VERification (FEVER) task (Thorne et al., 2018),[1] we begin by briefly describing the task setup. We are given a textual claim q, to be verified against a corpus comprised of a subset of Wikipedia. Each claim is associated with a three-way veracity label $v(q) \in \{\text{SUPPORTS}, \text{NOINFO}, \text{REFUTES}\}$ and a set of reference sentences $S(q)$ that provide support.[2] An example claim q, its label $v(q)$, and supporting evidence $S(q)$ are given in Figure 1.

The primary evaluation metric, FEVER score, is computed as the proportion of claims where the system has predicted the correct veracity label conditioned on also having retrieved a complete set of reference sentences. Most current systems adopt a three-stage approach to this task, comprising document retrieval, sentence selection, and label prediction. In this work, our contributions are focused on the second and third sub-tasks; for document retrieval, we simply augment current best practices with BM25 (Yang et al., 2017; Lin et al., 2021).

[1] Details of the FEVER sets are included in Appendix A.1.
[2] Each claim may have multiple different sets of reference sentences, any of which is sufficient as the support set.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 402–410
August 1–6, 2021. ©2021 Association for Computational Linguistics

Claim: The Rodney King riots took place in the most populous county in the USA.
Evidence 1 (`wiki/Los_Angeles_Riots`): The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.
Evidence 2 (`wiki/Los_Angeles_County`): Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.
Label: SUPPORTS

Figure 1: An example claim and its corresponding evidence and label from the FEVER dataset.

By construction, the veracity of each claim is determined by the (candidate) supporting sentences, taken together. One simple and popular approach to fact extraction and verification is to consider the veracity of the claim with respect to each candidate *independently* (i.e., classification), and then aggregate the evidence (Hanselowski et al., 2018; Zhou et al., 2019; Soleimani et al., 2019; Liu et al., 2020; Pradeep et al., 2021b). For convenience, we refer to these as "pointwise approaches", borrowing from the learning to rank literature (Li, 2011).

As an alternative, researchers have proposed approaches that consider multiple candidates at once to *jointly* arrive at a veracity prediction (Thorne et al., 2018; Nie et al., 2019a; Zhou et al., 2019; Stammbach and Neumann, 2019; Pradeep et al., 2021a). For convenience, we refer to these as "listwise approaches", also borrowing from the learning to rank literature (Li, 2011). Such listwise approaches have also been used for information aggregation in other NLP tasks such as question answering (Wang et al., 2018; Nie et al., 2019b). At a high level, this strategy suffers from a number of challenges, including data sparsity and a high level of sensitivity to noisy inputs. Following this thread of work, we adopt the listwise approach and improve it by training with a data augmentation technique that involves deliberately introducing noise into the training data to produce a more robust model.

3 Methods

Our work adopts a three-stage pipeline comprising document retrieval, sentence selection, and label prediction, which we detail in this section.

3.1 Document Retrieval

Given a claim q, our first step is to retrieve the top K most relevant documents $\hat{D}(q) = \{d_1, ..., d_K\}$. Since the corpus contains over 5.4M documents, we first perform document retrieval to narrow our search space. We leverage the Pyserini toolkit (Yang et al., 2017; Lin et al., 2021), which is based on the popular Lucene search engine, using the BM25 scoring function (Robertson and Zaragoza, 2009) to rank documents. Additional document retrieval details are described in Appendix A.3. We also incorporate document retrieval using the MediaWiki API, which has been shown in previous work to form a strong baseline (Hanselowski et al., 2018). We combine the results of the two methods by alternating through the two ranked lists of documents, skipping duplicates and keeping the top K unique documents.

3.2 Sentence Selection

Given a claim q and retrieved documents $\hat{D}(q)$, the next stage in the pipeline selects the top L most relevant evidence sentences $\hat{S}(q) = \{s_{k_1,i_1}, ..., s_{k_L,i_L}\}$, where $s_{k,i}$ is the i-th sentence from document d_k. Similar to how Soleimani et al. (2019) and Subramanian and Lee (2020) frame this stage as a semantic matching problem using BERT-based models, we use T5 to rank the similarities between the claim and the sentences in each document. Introduced by Nogueira et al. (2020), like Pradeep et al. (2021a), we use T5 (Raffel et al., 2020) as a pointwise reranker, which they dub monoT5. Empirically, T5 has been found to be more effective at ranking than BERT-based models across a wide variety of domains.

As a sequence-to-sequence model, ranking is performed using the following input template:

`Query:` q `Document:` $s_{k,i}$ `Relevant:`

where q and $s_{k,i}$ are the claim and evidence sentence, respectively. To provide a broader context and to resolve ambiguities, we prepend each sentence $s_{k,i}$ with the title of document d_k.

We fine-tune the model to generate the token "true" if $s_{k,i} \in S(q)$ and "false" otherwise. In terms of training data for fine-tuning, we use the gold evidence in the evidence sets in $S(q)$ for "true" samples, but for the "false" samples, we sample negatives from the sentences in $\hat{D}(q)$.

At inference time, we construct a candidate set comprised of sentences from each document in $\hat{D}(q)$ in its retrieved order. Using the same input

format, for each sentence, we probe the logits of the "true" and "false" tokens and apply the softmax function to produce a relevance probability score between 0 and 1; these scores are used to select the top L (= 5) sentences. For efficiency, instead of reranking all sentences in $\hat{D}(q)$, we take the first 200 sentences and only rerank this subset. Since there is an average of five non-empty sentences per document, we are roughly considering the top 40 documents from $\hat{D}(q)$.

On top of the basic reranking input template of Nogueira et al. (2020), we introduce a novel enhancement where we append any named entities found within the claim to the input of monoT5. The intuition here is to prompt monoT5 to promote sentences that come from documents with titles that are similar to those entities, which tend to contain information that is relevant to verifying the claims. During fine-tuning, we use the names of the documents that contain the gold evidence as entities, but during inference, we extract named entities from the claims using the named entity recognition (NER) module built into spaCy's `en_core_web_sm` model.[3] We append these entities, denoted as $e_1, ..., e_j$, to our monoT5 input template as follows:

`Query:` q `Document:` $s_{k,i}$ `Entity1:` e_1 $\cdots$ `Entity`j`:` e_j `Relevant:`

Additional details are described in Appendix A.3.

3.3 Label Prediction

Given claim q and evidence $\hat{S}(q)$, the final stage of the pipeline is to predict a veracity label $\hat{v}(q)$.

Pointwise Aggregation One common method in the literature for label prediction is to combine the claim and each evidence sentence individually as the inputs to some model and aggregate those model outputs to obtain a veracity prediction. With the sequence-to-sequence nature of T5, we achieve this by fine-tuning the model with samples of the following input sequence:

`query:` q `sentence:` $s_{k,i}$ `relevant:`

There are many different methods to aggregate the outputs: Soleimani et al. (2019) assumes NOINFO unless there are unanimous outputs of SUPPORTS or REFUTES, while Zhou et al. (2019) chooses the most frequently occurring label as well as attending over the outputs with the vector representations of each claim and evidence pair. Assume that the input sequence with evidence sentence $s_{k,i}$, after passing through T5 and applying the softmax function to the logits of the three classes, produces the probabilities $\Pr(\text{S} \mid q, s_{k,i})$ for SUPPORTS, $\Pr(\text{R} \mid q, s_{k,i})$ for REFUTES, and $\Pr(\text{N} \mid q, s_{k,i})$ for NOINFO. We experiment with two aggregation schemes that achieves the best results for us, which we denote by `sum` and `max`, as follows:

$$\texttt{sum} : \hat{v}(q) = \underset{l \in \{\text{S,R,N}\}}{\text{argmax}} \sum_{s_{k,i} \in \hat{S}(q)} \Pr(l \mid q, s_{k,i})$$

$$\texttt{max} : \hat{v}(q) = \underset{l \in \{\text{S,R,N}\}}{\text{argmax}} \max_{s_{k,i} \in \hat{S}(q)} \Pr(l \mid q, s_{k,i})$$

For fine-tuning, we use $S(q)$ as the evidence for SUPPORTS and REFUTES samples. Similar to sentence selection, for NOINFO samples, we sample negatives from the top predicted sentences from upstream, which in this case, is sentence selection, using the full reranked candidate list instead of just the top L sentences in $\hat{S}(q)$.

Listwise Concatenation Another common strategy for label prediction is to concatenate all L sentences into a single input to some model and have the model directly classify the claim and list of evidence $\hat{S}(q)$ as one of SUPPORTS, NOINFO, and REFUTES. Again, with T5, we use the following input sequence:

`query:` q `sentence1:` s_{k_1,i_1} $\cdots$ `sentence`L`:` s_{k_L,i_L} `relevant:`

To obtain fine-tuning training data, we use the same method as for pointwise aggregation.

Listwise Data Augmentation To make label prediction more tolerant to noisy evidence in the top L sentences, we fine-tune T5 with augmented, noisy evidence sets: this mimics the model during inference more closely as there usually exists some non-gold evidence in $\hat{S}(q)$. To accomplish this, instead of fine-tuning directly with the gold evidence sets $S(q)$, we fine-tune using $I(S(q))$, which "infuses" $S(q)$ with $\hat{S}(q)$. Specifically, we define the transformation I as:

- If $v(q) \in \{\text{SUPPORTS}, \text{REFUTES}\}$, we check if $S(q) \subseteq \hat{S}(q)$. For each $s \in S(q)$ such that $s \notin \hat{S}(q)$, we randomly select an index k of $\hat{S}(q)$ where $\hat{S}(q)[k] \notin S(q)$ and insert s at $\hat{S}(q)[k]$. This is repeated iteratively, and so $I(S(q))$ returns the resulting list of sentences $\hat{S}(q)$.
- If $v(q) = \text{NOINFO}$, $I(S(q)) = \hat{S}(q)$.

[3] https://spacy.io

Method		Dev		Test	
		LA (%)	FS (%)	LA (%)	FS (%)
(1a)	UNC (Nie et al., 2019a)	66.14	69.60	72.56	67.26
(1b)	Soleimani et al. (2019)	72.42	74.59	71.86	69.66
(1c)	HESM (Subramanian and Lee, 2020)	75.77	73.44	74.64	71.48
(1d)	CorefRoBERTa (Ye et al., 2020)	–	–	75.96	72.30
(1e)	GEAR (Zhou et al., 2019)	74.84	70.69	71.60	67.10
(1f)	DREAM (Zhong et al., 2020)	–	–	76.85	70.60
(1g)	`nudt_nlp`*	–	–	77.38	74.42
(1h)	`dominiks`*	–	–	76.60	74.27
(2a)	Oracle	–	94.74	–	–
(2b)	T5 w/ `sum` pointwise aggregation	63.19	59.45	–	–
(2c)	T5 w/ `max` pointwise aggregation	70.31	66.15	–	–
(2d)	T5 w/ listwise concatenation	70.66	67.18	–	–
(2e)	T5 w/ listwise data augmentation	**81.26**	**77.75**	**79.35**	**75.87**

Table 1: Label prediction results on the FEVER development set and blind test set. LA refers to label accuracy and FS refers to FEVER score. Other top submissions on the FEVER leaderboard at the time of our work are denoted with the symbol *.

Note that we use the same T5 input format as listwise concatenation. Training details for the label prediction stage can be found in Appendix A.3.

4 Results

We report the overall results of LisT5 on the FEVER development and blind test sets in Table 1, comparing the label prediction variations presented in Section 3.3. We also include the oracle FEVER score for our retrieved $\hat{S}(q)$ on line (2a). For reference, we compare LisT5 against several baselines and state-of-the-art techniques (drawn from the leaderboard) at the time of our work, shown in lines (1a)–(1h).

From the results in Table 1, it is clear that the different label prediction strategies lead to vastly different FEVER scores. The top-performing method, according to both label accuracy and FEVER score, is trained with augmented data in a listwise manner, found on line (2e). This run represents the state of the art atop the FEVER leaderboard at the time of our work. The other methods that fine-tune with only gold evidence data, found on lines (2b) to (2d), seem to trail by over 10 points. These results suggest the importance of training with augmented listwise evidence sets, which is presented in Section 3.3.

Contrary to the results reported in some papers, our concatenation methods consistently outperform corresponding aggregation methods: this suggests that T5 is able to capture inter-sentence semantics and use information from multiple, possibly diverse, pieces of evidence to come to veracity conclusions. Specifically, the T5 variant on line (2e) achieves 78.02%[4] (174/223) label accuracy on claims in the development set that require retrieving at least two pieces of evidence in conjunction to verify, which is close to our overall label accuracy of 81.26%. This finding suggests that T5 is capable of incorporating and corroborating the information contained in multiple pieces of evidence, which is one of the most common needed areas of improvement described in previous papers.

Table 2 compares the LisT5 sentence selection results of the monoT5 variations described in Section 3.2. We include some results from baselines, using recall at five as the primary sentence selection metric, which by definition is an upper-bound for the downstream FEVER score. We format the results for LisT5 as an ablation analysis focused on sentence selection. Line (2a) shows the results of the full monoT5 model with NER and fine-tuned on the FEVER dataset; monoT5 without NER features but fine-tuned on the FEVER dataset is shown on line (2b). Finally, we have zero-shot monoT5 on line (2c) to show the results of monoT5 without fine-tuning on the FEVER dataset, i.e., directly from the model checkpoints of Nogueira

[4]These only include claims where an entire gold evidence set is contained in the sentence selection output $\hat{S}(q)$.

Method		P@5 (%)	R@5 (%)	F1@5 (%)	MAP@5 (%)
(1a)	UNC (Nie et al., 2019a)	36.49	86.79	51.38	–
(1b)	Soleimani et al. (2019)	25.13	88.29	39.13	–
(1c)	HESM (Subramanian and Lee, 2020)	–	90.50	–	–
(1d)	GEAR (Zhou et al., 2019)	**40.60**	86.36	**55.23**	–
(1e)	DREAM (Zhong et al., 2020)	26.67	87.64	40.90	–
(2a)	monoT5 w/ NER (full model)	25.66	**90.54**	37.17	**85.62**
(2b)	monoT5 w/o NER (fine-tuned)	25.50	90.08	36.94	84.87
(2c)	monoT5 w/o NER (zero-shot)	22.70	85.39	33.86	76.87

Table 2: Comparison of sentence selection methods on the FEVER development set.

et al. (2020). We explain this in more detail in Appendix A.3. From these results, it is clear that monoT5 supplemented with named entities – on line (2a) – performs the best, achieving the highest recall and mean average precision, better than the other monoT5 variations or any of the baselines. It is worth noting that the full monoT5 model on line (2a) achieves 90.53 recall on the blind test set, consistent with the development set results.

While document retrieval is not our focus, our pipeline performs competitively compared to prior work and is further discussed in Appendix A.4.

5 Error Analysis

We randomly select 200 incorrectly predicted claims by LisT5 and summarize the most common issues, hoping to identify areas of improvement for future fact verification systems.

One common issue is failing to distinguish between similar but semantically different words or phrases. An example of this is the claim "Shane McMahon officially retired on the first day of 2010" to which our document retrieval and sentence selection stages retrieve the sentence "In 2009, McMahon announced his resignation from WWE which went into effect January 1, 2010". Here, retirement and resignation are semantically similar words that both describe individuals leaving their positions. These similarities may have been learned by the pretrained transformer, but it is not always the case that the words imply one another, leading to an incorrect prediction for this claim.

Another frequent issue is incorrectly labelled claims in the FEVER dataset, often due to missing evidence in $S(q)$. An example of this is the claim "Mickey Rourke appeared in a sequel" to which our document retrieval and sentence selection stages retrieve the sentence "Since then, Rourke has appeared in several commercially successful films including the 2010 films Iron Man 2 and The Expendables and the 2011 film Immortals". However, the claim was labelled NOINFO in the dataset, which is incorrect due to Iron Man 2 indeed being a sequel. In short, we are bumping into data quality issues in the annotations themselves.

6 Conclusion

In this paper, we present the LisT5 framework for automated fact verification. LisT5 consists of a three-stage pipeline – document retrieval, sentence selection, and label prediction. For document retrieval, we combine two strong document retrieval baselines. For sentence selection, we fine-tune a T5 model as a reranker with named entities provided as additional features. For label prediction, we present evidence in a listwise manner to a T5 model, trained on augmented data. Our experimental results indicate that LisT5 achieves the state of the art on the FEVER task, which we attribute to the framework's ability to reason jointly over multiple pieces of evidence.

Acknowledgments

This research was supported in part by the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Waterloo–Huawei Joint Innovation Laboratory. Additionally, we would like to thank Google's TensorFlow Research Cloud (TFRC) for access to Cloud TPUs.

References

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3490–3496, Hong Kong, China.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv:1611.09268v3*.

Tuhin Chakrabarty, Tariq Alhindi, and Smaranda Muresan. 2018. Robust document retrieval and individual evidence modeling for fact extraction and verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 127–131, Brussels, Belgium.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-Athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium.

Hang Li. 2011. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019a. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6859–6866.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019b. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2553–2566, Hong Kong, China.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021a. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021b. Vera: Prediction techniques for reducing harmful misinformation in consumer health search. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*.

Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Nogueira, and Jimmy Lin. 2020. H_2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020)*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021c. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv:2101.05667*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Amir Soleimani, Christof Monz, and Marcel Worring. 2019. BERT for evidence retrieval and claim verification. *arXiv:1910.02655*.

Dominik Stammbach and Guenter Neumann. 2019. Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 105–109, Hong Kong, China.

Shyam Subramanian and Kyumin Lee. 2020. Hierarchical Evidence Set Modeling for automated fact extraction and verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7798–7809, Online.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence aggregation for answer re-ranking in open-domain question answering. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, pages 1253–1256, Tokyo, Japan.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. UCL machine reading group: Four factor framework for fact finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102, Brussels, Belgium.

Edwin Zhang, Nikhil Gupta, Raphael Tang, Xiao Han, Ronak Pradeep, Kuang Lu, Yue Zhang, Rodrigo Nogueira, Kyunghyun Cho, Hui Fang, and Jimmy Lin. 2020. Covidex: Neural ranking models and keyword search infrastructure for the COVID-19 Open Research Dataset. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 31–41, Online.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy.

A Appendix

A.1 FEVER Dataset

The dataset used for training and evaluating our fact verification system is FEVER (Thorne et al., 2018), a large-scale dataset consisting of 185K claims with evidence taken from Wikipedia. We include the label distribution of the dataset across its training, development, and blind test set in Table 3.

Split	SUPPORTS	REFUTES	NOINFO
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 3: Label distribution of the FEVER dataset.

A.2 Baseline Details

As discussed in Section 3, most fact verification systems, especially for the FEVER task, consist of a three-stage pipeline similar to the one used in LisT5. The stages are as follows:

Document Retrieval Many systems use the document retrieval component of DrQA (Chen et al., 2017a), which performs retrieval with TF-IDF feature vectors along with bigram features. Some other systems leverage external search APIs, such as Hanselowski et al. (2018), who use the MediaWiki API, Wikipedia's own search engine, and Chakrabarty et al. (2018), who use the Google Search API.

Sentence Selection When the FEVER task was introduced in 2018, many of the initial top-scoring systems (Nie et al., 2019a; Hanselowski et al., 2018) employed variations of the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017b), which consists of Bidirectional Long Short-Term Memory networks (BiLSTMs) (Schuster and Paliwal, 1997) as its primary building block. However, with the advent of the transformer architecture (Vaswani et al., 2017), most systems today (Soleimani et al., 2019; Subramanian and Lee, 2020) use transformers to perform semantic matching between the claim and each candidate sentence.

Label Prediction Framing the problem as that of natural language inference (NLI), Nie et al. (2019a), Yoneda et al. (2018), and Hanselowski et al. (2018) again use variations of ESIM for label prediction. Similar to the sentence selection stage, many recent systems use transformers for this stage as well. However, there has also been active research into graph-based models for knowledge aggregation by modelling evidence sentences as nodes in a graph (Zhou et al., 2019; Liu et al., 2020; Zhong et al., 2020).

A.3 Implementation and Training Details

Document Retrieval We retrieve with BM25 using the parameters $k_1 = 0.6$ and $b = 0.5$. These parameters are tuned by running a grid search over parameter values in 0.1 increments over a subset of the training set.

Sentence Selection Whenever we fine-tune monoT5, we use the T5-3B variant, which as its name suggests, contains three billion parameters. We fine-tune the model with batch size 128 over one epoch, using the configurations prescribed by Raffel et al. (2020), except that we use learning rate 0.0001 instead of 0.001. While training, we save checkpoints at evenly spaced iteration intervals, usually around 1000 iterations per checkpoint depending on the size of the training data. Thus, whenever we report the results of a model, we use the results of the best performing checkpoint on the FEVER development set. We fine-tune on TPU v3-8 nodes on the Google Cloud Platform, which takes around 24 hours.

Note that we first fine-tune a pretrained T5 model on the MS MARCO passage dataset (Bajaj et al., 2018) for 10000 iterations, following best practices reported in previous work (Akkalyoncu Yilmaz et al., 2019; Nogueira et al., 2020; Zhang et al., 2020; Pradeep et al., 2020, 2021c,b), which has shown that this leads to improved effectiveness. This procedure also gives us a zero-shot setting for fact verification, which we experiment with before fine-tuning on the FEVER dataset directly.

In our experiments, we note that negative sampling sentences from highly-ranked documents in $\hat{D}(q)$ leads to poorly performing models. This may be due to false negatives in the data, where some claims are labelled as NOINFO but are actually verifiable, with relevant evidence retrieved by our document retrieval stage. To avoid negative sampling such false negative evidence, we negative sample sentences ranked between 50 and 200.

Label Prediction Again, we use the T5-3B variant as the model for label prediction. We use similar settings for fine-tuning T5 as before for monoT5, except that we use the default learning rate 0.001.

Method	R@1000 (%)
MediaWiki API	89.56
Anserini	94.76
Anserini + MediaWiki API	**96.87**

Table 4: Comparison of document retrieval methods on the FEVER development set. The code for retrieval using the MediaWiki API is courtesy of Hanselowski et al. (2018).

We also fine-tune on TPU v3-8 nodes on the Google Cloud Platform, which takes around 8 hours.

To avoid similar negative sampling issues encountered in fine-tuning models for sentence selection, we sample from sentences ranked between 10 and 25 here.

A.4 Document Retrieval Results

We report the importance of combining the two document retrieval methods described in Section 3.1 by comparing their recall at rank 1000 in Table 4. These figures show that combining the two techniques results in being only a few points away from perfectly retrieving all relevant documents.

DefSent: Sentence Embeddings using Definition Sentences

Hayato Tsukagoshi **Ryohei Sasano** **Koichi Takeda**
Graduate School of Informatics, Nagoya University
tsukagoshi.hayato@e.mbox.nagoya-u.ac.jp,
{sasano,takedasu}@i.nagoya-u.ac.jp

Abstract

Sentence embedding methods using natural language inference (NLI) datasets have been successfully applied to various tasks. However, these methods are only available for limited languages due to relying heavily on the large NLI datasets. In this paper, we propose DefSent, a sentence embedding method that uses definition sentences from a word dictionary, which performs comparably on unsupervised semantics textual similarity (STS) tasks and slightly better on SentEval tasks than conventional methods. Since dictionaries are available for many languages, DefSent is more broadly applicable than methods using NLI datasets without constructing additional datasets. We demonstrate that DefSent performs comparably on unsupervised semantics textual similarity (STS) tasks and slightly better on SentEval tasks to the methods using large NLI datasets. Our code is publicly available at https://github.com/hpprc/defsent.

1 Introduction

Sentence embeddings represent sentences as dense vectors in a low dimensional space. Recently, sentence embedding methods using natural language inference (NLI) datasets have been successfully applied to various tasks, including semantic textual similarity (STS) tasks. However, these methods are only available for limited languages due to relying heavily on the large NLI datasets. In this paper, we propose DefSent, a sentence embedding method that uses definition sentences from a word dictionary. Since dictionaries are available for many languages, DefSent is more broadly applicable than the methods using NLI datasets without constructing additional datasets.

Defsent is similar to the model proposed by Hill et al. (2016) in that it generates sentence embeddings so that the embeddings of a definition sentence and the word it represents are similar. However, while Hill et al. (2016)'s model is based on recurrent neural network language models, DefSent is based on pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), with a fine-tuning mechanism as well as Sentence-BERT (Reimers and Gurevych, 2019). Sentence-BERT is one of the state-of-the-art sentence embedding models, which is based on pre-trained language models that are fine-tuned on NLI datasets. Overviews of Sentence-BERT and DefSent are depicted on Figure 1.

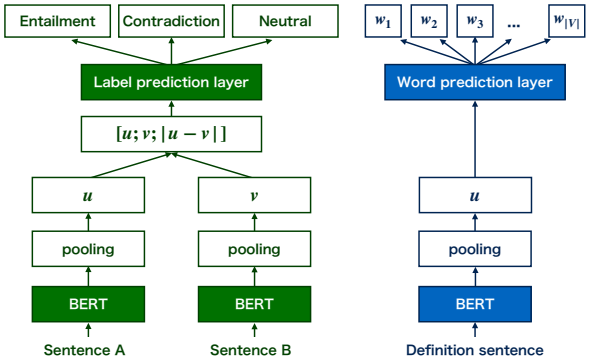


Figure 1: Sentence-BERT (left) and DefSent (right).

2 Sentence Embedding Methods

In this section, we introduce BERT, RoBERTa, and Sentence-BERT, followed by a description of DefSent, our proposed sentence embedding method.

2.1 BERT and RoBERTa

BERT is a pre-trained language model based on the Transformer architecture (Vaswani et al., 2017). Utilizing masked language modeling and next sentence prediction, BERT acquires linguistic knowledge and outputs contextualized word embeddings. In masked language modeling, a specific proportion of input tokens is replaced with a special token [MASK], and the model is trained to predict these

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 411–418
August 1–6, 2021. ©2021 Association for Computational Linguistics

masked tokens. Next sentence prediction is a task to predict whether two sentences connected by a sentence separator token [SEP] are consecutive sentences in the original text data. BERT uses the output embedding of the unique token [CLS] at the beginning of each such sentence for prediction.

RoBERTa has the same structure as BERT. It attempts to improve BERT by removing the next sentence prediction from pre-training objectives and increasing the data size and batch size. While both Sentence-BERT and DefSent are applicable to BERT and RoBERTa, we use BERT for the explanations in this paper.

2.2 Sentence-BERT

Conneau et al. (2017) proposed InferSent, a sentence encoder based on a Siamese network structure. InferSent trains the sentence encoder such that similar sentences are distributed close to each other in the semantic space. Reimers and Gurevych (2019) proposed Sentence-BERT, which also uses a Siamese network to create BERT-based sentence embeddings. An overview of Sentence-BERT is depicted on the left side of Figure 1. Sentence-BERT first inputs the sentences to BERT and then constructs a sentence embedding from the output contextualized word embeddings by pooling. They utilize the following three types of pooling strategy.

CLS Using the [CLS] token embedding.; When using RoBERTa, since the [CLS] token does not exist, the beginning-of-sentence token <s> is used as an alternative.

Mean Using the mean of the contextualized embeddings of all words in a sentence.

Max Using the max-over-time of the contextualized embeddings of all words in a sentence.

Let u and v be the sentence embeddings for each of the sentence pairs obtained by pooling. Then compose a vector $[u; v; |u - v|]$ and feed it to the label prediction layer, which has the same number of output dimensions as the number of classes. For fine-tuning, Reimers and Gurevych uses the SNLI dataset (Bowman et al., 2015) and the Multi-Genre NLI dataset (Williams et al., 2018), which together contain about one million sentences.

2.3 DefSent

Since they have the same meaning, we focus on the relationship between a definition sentence and the word it represents. To learn how to embed sentences in the semantic vector space, we train the sentence embedding model by predicting the word from definitions. An overview of DefSent is depicted on the right side of Figure 1. We call the layer that predicts the original token from the [MASK] embeddings used in the masked language modeling during BERT pre-training a word prediction layer. Also, we use w_k to denote the word corresponding to a given definition sentence X_k.

DefSent inputs the definition sentence X_k to BERT and derives the sentence embedding u by pooling the output embeddings. As in Sentence-BERT, three types of pooling strategy are used: CLS, Mean, and Max. Then, the derived sentence embedding u is input to the word prediction layer to obtain the probability $P(w_k|X_k)$. We use cross-entropy loss as a loss function and fine-tune BERT to maximize $P(w_k|X_k)$.

In DefSent, the parameters of the word prediction layer are fixed. This setting allows us to fine-tune models without training an additional classifier, as is the case with both InferSent and Sentence-BERT. Additionally, since our method uses a word prediction layer that has been pre-trained in masked language modeling, the sentence embedding u is expected to be similar to the contextualized word embedding of w_k when w_k appears as the same meaning as X_k.

3 Word Prediction Experiment

To evaluate how well DefSent can predict words from sentence embeddings, we conducted an experiment to predict a word from its definition.

3.1 Dataset

DefSent requires pairs of a word and its definition sentence. We extracted these from the Oxford Dictionary dataset used by Ishiwatari et al. (2019). Each entry in the dataset consists of a word and its definition sentence, and a word can have multiple definitions. We split this dataset into train, dev, and test sets in the ratio of 8:1:1 word by word to evaluate how well the model can embed unseen definitions of unseen words. It is worth noting that since DefSent utilizes the pre-trained word prediction layer of BERT and RoBERTa, it is impossible to obtain probabilities for out-of-vocabulary (OOV) words. Therefore, we cannot calculate losses of these OOV words in a straightforward way.[1] In our

[1] Although we could substitute the mean of subwords as OOV word embeddings, we opted to filter out OOV words for

experiments, we only use words and their respective definitions in the dataset, as contained by the model vocabulary. The statistics of the datasets are listed in Table 1.

3.2 Settings

We used the following pre-trained models: BERT-base (bert-base-uncased), BERT-large (bert-large-uncased), RoBERTa-base (roberta-base), and RoBERTa-large (roberta-large) from Transformers (Wolf et al., 2020). The batch size was 16, a fine-tuning epoch size was 1, the optimizer was Adam (Kingma and Ba, 2015), and we set a linear learning rate warm-up over 10% of the training data. For each respective model and pooling strategy, the learning rate was chosen based on the highest recorded Mean Reciprocal Rank (MRR) for the dev set in the range of $2^x \times 10^{-6}, x \in \{0, 0.5, 1, ..., 7\}$. We conducted experiments with ten different random seeds, and their mean was used as the evaluation score. Top-k accuracy (the percentage of correct answers within the first, third, and tenth positions) and MRR were calculated from the output word probabilities when a definition sentence was fed into the model. Also, we evaluated the performance of BERT-base without fine-tuning for comparison.

3.3 Results

Table 2 shows the experimental results.[2] `Max` was the best pooling strategy for BERT-base without fine-tuning, but its top-1 accuracy was extremely low at 0.0157. This indicates that it is not adequate for predicting words from definitions without fine-tuning. DefSent performed higher for larger models. In the case of BERT, `CLS` was the best pooling strategy for both base and large models. `CLS` was also the best pooling strategy for RoBERTa-base but `Mean` was the best for RoBERTa-large.

4 Extrinsic Evaluations

Next, to evaluate the general quality of the constructed sentence embedding, we conducted evaluations on semantic textual similarity (STS) tasks and SentEval tasks (Conneau and Kiela, 2018).

simplicity and intuitiveness.

[2]We report the fine-tuning time and computing infrastructure in Appendix A, and report the learning rate, means, and standard deviations on the word prediction experiment in Appendix B. We also show the actual predicted words when definition sentences and other sentences are given as inputs in Appendices C and D, respectively.

All	Words	Definitions	Avg. length
Train	29,413	97,759	9.921
Dev	3,677	12,127	9.874
Test	3,677	12,433	9.846
In BERT vocab.	**Words**	**Definitions**	**Avg. length**
Train	7,732	54,142	9.531
Dev	936	6,544	9.512
Test	979	6,930	9.551
In RoBERTa vocab.	**Words**	**Definitions**	**Avg. length**
Train	7,269	53,935	9.376
Dev	901	6,625	9.372
Test	925	6,945	9.410

Table 1: Statistics of datasets.

Model	Pooling	MRR	Top1	Top3	Top10
BERT-base (no fine-tuning)	`CLS`	.0009	.0000	.0000	.0000
	`Mean`	.0132	.0001	.0043	.0242
	`Max`	.0327	.0157	.0320	.0626
BERT-base	`CLS`	.3200	.2079	.3670	.5418
	`Mean`	.3091	.1972	.3524	.5356
	`Max`	.2939	.1840	.3350	.5207
BERT-large	`CLS`	**.3587**	**.2388**	**.4139**	**.6011**
	`Mean`	.3286	.2091	.3792	.5723
	`Max`	.2925	.1814	.3356	.5194
RoBERTa-base	`CLS`	.3436	.2241	.3983	.5836
	`Mean`	.3365	.2170	.3906	.5783
	`Max`	.3072	.1941	.3523	.5386
RoBERTa-large	`CLS`	.3863	.2611	.4460	.6364
	`Mean`	**.3995**	**.2699**	**.4634**	**.6599**
	`Max`	.3175	.2015	.3646	.5543

Table 2: Results of word prediction experiments.

4.1 Settings

We compared the performance of DefSent with several existing sentence embedding methods including InferSent (Conneau et al., 2017), Universal Sentence Encoder (Cer et al., 2018), and Sentence-BERT (Reimers and Gurevych, 2019). For the pooling strategies, we used the strategy that achieved the highest MRR in the word prediction task for each pre-trained model.[3] The performance of the existing methods was taken from Reimers and Gurevych (2019).

4.2 Semantic textual similarity tasks

We evaluated DefSent on unsupervised STS tasks. In these tasks, we compute semantic similarities of given sentence pairs and calculate Spearman's rank correlation ρ between similarities and gold scores of sentence similarities. In the unsupervised setting, none of the models are optimized on the STS datasets. Instead, the similarities of the given sentence embeddings are calculated using common similarity measures such as negative Manhattan distance, negative Euclidean distance, and cosine-similarity. In this study, we used cosine-similarity.

[3]We report the means and standard deviations on the unsupervised STS tasks and SentEval tasks for each respective model and pooling strategy in Appendices E and F.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Avg. GloVe embeddings (Pennington et al., 2014)	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - Glove (Conneau et al., 2017)	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder (Cer et al., 2018)	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
Sentence-BERT-base (Mean)	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
Sentence-BERT-large (Mean)	72.27	78.46	**74.90**	**80.99**	76.25	**79.23**	73.75	76.55
Sentence-RoBERTa-base (Mean)	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
Sentence-RoBERTa-large (Mean)	**74.53**	77.00	73.18	81.85	76.82	79.10	74.29	**76.68**
DefSent-BERT-base (CLS)	67.56	79.86	69.52	76.83	76.61	75.57	73.05	74.14
DefSent-BERT-large (CLS)	66.22	**82.07**	71.48	79.34	75.38	73.46	74.30	74.61
DefSent-RoBERTa-base (CLS)	65.55	80.84	71.87	78.77	**79.29**	78.13	74.92	75.62
DefSent-RoBERTa-large (Mean)	58.36	76.24	69.55	73.15	76.90	78.53	73.81	72.36

Table 3: Spearman's rank correlation $\rho \times 100$ between cosine similarities of sentence embeddings and human ratings. STS-B denotes STS Benchmark, and SICK-R denotes SICK-Relatedness.

Model	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
Avg. GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.80	69.45	84.94
BERT CLS-vector	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
InferSent - GloVe	81.57	86.54	92.50	90.38	84.18	88.20	75.77	85.59
Universal Sentence Encoder	80.09	85.19	93.98	86.70	86.38	**93.20**	70.14	85.10
Sentence-BERT-base (Mean)	83.64	89.43	94.39	89.86	88.96	89.60	**76.00**	87.41
Sentence-BERT-large (Mean)	84.88	90.07	94.52	90.33	90.66	87.40	75.94	87.69
DefSent-BERT-base (CLS)	80.94	87.57	94.59	89.98	85.78	89.73	73.82	86.06
DefSent-BERT-large (CLS)	85.79	90.54	**95.58**	90.15	**91.17**	90.47	73.74	88.20
DefSent-RoBERTa-base (CLS)	83.94	90.44	94.05	90.70	89.16	90.80	75.52	87.80
DefSent-RoBERTa-large (Mean)	**86.47**	**91.53**	95.02	**91.15**	90.77	92.33	73.91	**88.74**

Table 4: Accuracy (%) for each task in SentEval.

We performed experiments on unsupervised STS tasks using the STS12-16 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017), and SICK-Relatedness (Marelli et al., 2014) datasets. These datasets contain sentence pairs and their similarity scores, which is a real number from 0 to 5 assigned by human evaluations. Experiments were conducted with ten different random seeds, and the mean was used as the evaluation score.

Table 3 shows the experimental results. Although the training data size used in DefSent was only about 5% that of Sentence-BERT, DefSent-BERT-base and DefSent-RoBERTa-base performed comparably to Sentence-BERT-base and Sentence-RoBERTa-base. In particular, DefSent-RoBERTa models showed high performance in the STS Benchmark.

4.3 SentEval

SentEval (Conneau and Kiela, 2018) is a popular toolkit for evaluating the quality of universal sentence embeddings that aggregates various tasks, including binary and multi-class classification, natural language inference, and sentence similarity. For the SentEval evaluations, we trained a logistic regression classifier using sentence embeddings as input features to evaluate the extent to which each sentence embedding contained the important information for each task. We used the same tasks and settings as Reimers and Gurevych (2019) and performed a 10-fold cross-validation. We conducted experiments with three different random seeds, and the mean was used as the evaluation score.

Table 4 shows the results.[4] DefSent-RoBERTa-large achieved the best average score among all models. Also, increasing the model size improved the performance consistently. The performances of DefSent-BERT-large, DefSent-RoBERTa-base, and DefSent-RoBERTa-large were better than the performances of Sentence-BERT-based methods. These results indicate that DefSent embeds useful information that can be applied to various tasks.

5 Conclusion

In this paper, we proposed DefSent, a new sentence embedding method using a dictionary, and demonstrated its effectiveness through a series of experiments. Its performance was comparable to or even slightly better than existing methods using

[4] Reimers and Gurevych (2019) reported that there were minor difference from Sentence-BERT, so we omitted the results of Sentence-RoBERTa.

large NLI datasets. DefSent is based on dictionaries developed for many languages, so it does not require new language resources when applied to other languages. Since the model is trained with the same word prediction process as the masked language modeling, sentence embeddings derived by DefSent are expected to be similar to contextualized word embeddings of a word when it appears with the same meaning as the definition.

In future work, we will evaluate the performance of DefSent when it is applied to languages other than English and when it is applied to a broader range of downstream tasks, such as document classification tasks. We will also analyze the relationship between the sentence embeddings by DefSent and the contextualized word embeddings in the semantic vector space and investigate how model architecture and size influence the embeddings.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 21H04901.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Semantic Evaluation (SemEval)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 32–43.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval)*, pages 1–14.

Daniel Matthew Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, C. Tar, Yun-Hsuan Sung, B. Strope, and R. Kurzweil. 2018. Universal Sentence Encoder. *arXiv:1803.11175*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, pages 1699–1704.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to Understand Phrases by Embedding the Dictionary. In *Transactions of the Association for Computational Linguistics (TACL)*, pages 17–30.

Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to Describe Unknown Phrases with Local and Global Contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 3467–3476.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 216–223.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45.

A Average Runtime and Computing Infrastructure

Fine-tuning for DefSent-BERT-base and DefSent-RoBERTa-base took about 5 minutes on a single NVIDIA GeForce GTX 1080 Ti. Fine-tuning for DefSent-BERT-large and DefSent-RoBERTa-large took about 15 minutes on a single Quadro GV100.

B Full Results of the Word Prediction Experiment

Table 5 shows the experimental results on the word prediction experiment for each model and pooling strategy with learning rate.

C Predictions for definition sentences

Table 6 shows the predicted words when the embeddings of definition sentences are input. We used BERT-large as a model and `CLS` as a pooling strategy for the experiment. For prediction, sentences were first input into the model to obtain sentence embeddings. Then the sentence embeddings were input into the pre-trained word prediction layer to obtain word probabilities. We show the top five words with the highest probability.

D Predictions for sentences other than definition sentences

Table 7 shows the predicted words when the embeddings of sentences other than definition sentences are input. We used BERT-large as a model and `CLS` as a pooling strategy for the experiment. The evaluation procedure is the same as for Appendix C.

E Full Results of the STS Evaluation

Table 8 shows the experimental results on STS tasks for each model and pooling strategy.

F Full Results of the SentEval Evaluation

Table 9 shows the experimental results on SentEval tasks for each model and pooling strategy.

Model	Pooling	Learning rate	MRR	Top1	Top3	Top10
BERT-base	`CLS`	$2^{2.5} \times 10^{-6}$	.3200$\pm$.0020	.2079$\pm$.0021	.3670$\pm$.0029	.5418$\pm$.0022
	`Mean`	$2^{3.5} \times 10^{-6}$	.3091$\pm$.0021	.1972$\pm$.0030	.3524$\pm$.0038	.5356$\pm$.0029
	`Max`	$2^{3.5} \times 10^{-6}$	.2939$\pm$.0021	.1840$\pm$.0026	.3350$\pm$.0023	.5207$\pm$.0045
BERT-large	`CLS`	$2^{2.5} \times 10^{-6}$	.3587$\pm$.0043	.2388$\pm$.0047	.4139$\pm$.0059	.6011$\pm$.0054
	`Mean`	$2^{3.5} \times 10^{-6}$	.3286$\pm$.0044	.2091$\pm$.0045	.3792$\pm$.0055	.5723$\pm$.0072
	`Max`	$2^{3.0} \times 10^{-6}$	.2925$\pm$.0138	.1814$\pm$.0113	.3356$\pm$.0172	.5194$\pm$.0181
RoBERTa-base	`CLS`	$2^{2.5} \times 10^{-6}$	.3436$\pm$.0016	.2241$\pm$.0016	.3983$\pm$.0027	.5836$\pm$.0017
	`Mean`	$2^{3.0} \times 10^{-6}$	.3365$\pm$.0017	.2170$\pm$.0014	.3906$\pm$.0029	.5783$\pm$.0022
	`Max`	$2^{2.0} \times 10^{-6}$	.3072$\pm$.0037	.1941$\pm$.0039	.3523$\pm$.0050	.5386$\pm$.0064
RoBERTa-large	`CLS`	$2^{2.0} \times 10^{-6}$	.3863$\pm$.0040	.2611$\pm$.0045	.4460$\pm$.0044	.6364$\pm$.0041
	`Mean`	$2^{2.0} \times 10^{-6}$	.3995$\pm$.0041	.2699$\pm$.0053	.4634$\pm$.0042	.6599$\pm$.0036
	`Max`	$2^{2.5} \times 10^{-6}$	.3175$\pm$.0069	.2015$\pm$.0054	.3646$\pm$.0087	.5543$\pm$.0092

Table 5: MRR, top-1, top-3, and top-10 accuracy on the word prediction experiment. The scores are the mean and standard deviation of 10 evaluations with different random seeds.

Word	Definition	Predictions (1st, 2nd, 3rd)		
cost	be expensive for (someone)	**cost**	charge	pay
preserve	prevent (food) from rotting	**preserve**	keep	spoil
good	that which is pleasing or valuable or useful	**good**	pleasing	pleasure
linux	an open-source operating system modelled on unix.	**linux**	unix	gnu
pile	place or lay as if in a pile	**pile**	stack	heap
weird	very strange; bizarre	**weird**	strange	bizarre
sale	the general activity of selling	selling	**sale**	retail
satellite	a celestial body orbiting the earth or another planet.	planet	**satellite**	orbit
logic	the quality of being justifiable by reason	reason	justice	certainty
custom	a thing that one does habitually	habit	routine	ritual
chief	a person who is in charge	leader	boss	master
nirvana	an ideal or idyllic state or place	paradise	dream	ideal

Table 6: Predicted words when the embeddings of definition sentences are input. The first two columns represent words and their defining sentences, and the third to fifth columns represent the top three predicted words. Correctly predicted words shown in **bold**.

Input	Predictions (1st, 2nd, 3rd, 4th, 5th)				
royal man	king	royal	prince	noble	knight
royal woman	queen	princess	royal	regal	sovereign
royal boy	boy	prince	royal	king	baby
royal girl	princess	queen	lady	royal	belle
good	fine	good	great	right	solid
bad	bad	dirty	awful	ugly	nasty
not good	bad	poor	wrong	awful	terrible
not bad	okay	fair	good	fine	ok
Star wars	jedi	star	trek	galaxy	saga
Star wars in America	jedi	western	fan	hollywood	movie
Star wars in Europe	trek	space	adventure	cinema	fantas
Star wars in Japan	godzilla	anime	gundam	jedi	manga
captain america	marvel	hero	thor	superhero	hulk

Table 7: Predicted words when the embeddings of sentences other than definition sentences are input.

Model	Pooling	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT-base	CLS	67.56±0.26	79.86±0.25	69.52±0.39	76.83±0.32	76.61±0.33	75.57±0.37	73.05±0.32	74.14±0.25
	Mean	67.30±0.44	81.96±0.24	71.92±0.28	77.68±0.47	76.71±0.48	76.90±0.40	73.28±0.30	75.11±0.21
	Max	64.61±0.87	82.06±0.21	72.43±0.31	76.56±0.74	75.61±0.43	76.61±0.52	72.15±0.46	74.29±0.33
BERT-large	CLS	66.22±0.79	82.07±0.39	71.48±0.33	79.34±0.44	75.38±0.60	73.46±0.45	74.30±0.50	74.61±0.41
	Mean	64.18±0.96	82.76±0.42	73.14±0.32	79.66±0.92	77.93±0.78	77.89±0.89	73.98±0.46	75.65±0.53
	Max	58.94±1.06	81.03±0.66	71.34±0.88	76.23±1.83	76.07±0.56	75.75±0.70	71.69±0.74	73.01±0.74
RoBERTa-base	CLS	65.55±0.89	80.84±0.26	71.87±0.39	78.77±0.70	79.29±0.27	78.13±0.61	74.92±0.18	75.62±0.38
	Mean	60.78±1.41	77.17±0.60	69.71±0.73	75.13±1.00	77.75±0.38	76.52±0.63	74.10±0.45	73.02±0.63
	Max	63.85±0.86	78.55±0.90	71.19±0.86	76.55±1.12	77.86±0.59	78.02±0.77	73.97±0.46	74.28±0.62
RoBERTa-large	CLS	63.84±1.34	77.33±2.53	68.64±1.34	72.86±1.96	77.13±1.32	78.32±1.08	74.14±1.31	73.18±1.20
	Mean	58.36±1.16	76.24±0.87	69.55±0.85	73.15±1.32	76.90±0.94	78.53±0.54	73.81±0.88	72.36±0.73
	Max	62.89±1.42	77.99±1.88	69.83±1.66	75.60±1.51	79.63±0.60	79.34±0.48	74.04±0.84	74.19±0.88

Table 8: Spearman's rank correlation $\rho \times 100$ between the cosine similarities of the sentence embeddings and the human ratings for each model and pooling strategy. The scores are the mean and standard deviation of 10 evaluations with different random seeds.

Model	Pooling	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
BERT-base	CLS	80.94±0.08	87.57±0.12	94.59±0.09	89.98±0.04	85.78±1.14	89.73±0.76	73.82±0.19	86.06±0.28
	Mean	81.84±0.17	88.20±0.04	94.82±0.12	89.94±0.12	86.49±0.20	89.73±0.31	75.32±0.78	86.62±0.18
	Max	80.74±0.16	88.00±0.09	94.32±0.07	89.92±0.25	85.03±0.09	89.13±0.50	74.11±0.49	85.89±0.02
BERT-large	CLS	85.79±0.19	90.54±0.26	95.58±0.14	90.15±0.04	91.17±0.06	90.47±0.95	73.74±0.61	88.20±0.07
	Mean	84.05±0.25	89.50±0.24	95.21±0.12	90.19±0.36	89.44±0.14	88.60±0.87	73.99±0.90	87.28±0.05
	Max	83.48±0.30	89.04±0.37	94.55±0.09	89.88±0.17	87.50±0.26	90.87±1.30	74.28±1.27	87.09±0.27
RoBERTa-base	CLS	83.94±0.30	90.44±0.49	94.05±0.06	90.70±0.17	89.16±0.22	90.80±0.35	75.52±0.42	87.80±0.20
	Mean	84.88±0.21	91.09±0.01	94.60±0.10	90.69±0.07	89.73±0.54	93.13±0.12	77.22±0.46	88.76±0.08
	Max	83.98±0.03	90.78±0.24	93.96±0.07	90.63±0.11	90.05±0.06	93.60±0.72	77.80±0.32	88.69±0.12
RoBERTa-large	CLS	85.63±0.27	90.74±0.15	94.53±0.14	91.20±0.11	90.08±0.59	93.53±0.76	72.66±1.73	88.34±0.28
	Mean	86.47±0.29	91.53±0.06	95.02±0.08	91.15±0.07	90.77±0.34	92.33±0.64	73.91±0.96	88.74±0.12
	Max	85.60±0.26	90.73±0.70	94.21±0.65	91.09±0.32	90.65±0.37	91.53±1.70	76.15±0.33	88.56±0.57

Table 9: The percentage of correct answers (%) for each task of SentEval. The scores are the mean and standard deviation of three evaluations with different random seeds.

Discrete Cosine Transform as Universal Sentence Encoder

Nada Almarwani [1,2] and **Mona Diab** [1,3]
[1] Dep. of Computer Science, The George Washington University
[2] Dep. of Computer Science, College of Computer Science and Engineering, Taibah University
[3] Facebook AI Research
nmarwani@taibah.edu.sa, mdiab@fb.com

Abstract

Modern sentence encoders are used to generate dense vector representations that capture the underlying linguistic characteristics for a sequence of words, including phrases, sentences, or paragraphs. These kinds of representations are ideal for training a classifier for an end task such as sentiment analysis, question answering and text classification. Different models have been proposed to efficiently generate general purpose sentence representations to be used in pretraining protocols. While averaging is the most commonly used efficient sentence encoder, Discrete Cosine Transform (DCT) was recently proposed as an alternative that captures the underlying syntactic characteristics of a given text without compromising practical efficiency compared to averaging. However, as with most other sentence encoders, the DCT sentence encoder was only evaluated in English. To this end, we utilize DCT encoder to generate universal sentence representation for different languages such as German, French, Spanish and Russian. The experimental results clearly show the superior effectiveness of DCT encoding in which consistent performance improvements are achieved over strong baselines on multiple standardized datasets.

1 Introduction

Recently, a number of sentence encoding representations have been developed to accommodate the need of sentence-level understanding; some of these models are discussed in (Hill et al., 2016; Logeswaran and Lee, 2018; Conneau et al., 2017), yet most of these representations have focused on English only.

To generate sentence representations in different languages, the most obvious solution is to train monolingual sentence encoders for each language. However, training a heavily parameterized monolingual sentence encoder for every language is inefficient and computationally expensive, let alone the impact on the environment. Thus, utilizing a non-parameterized model with ready-to-use word embeddings is an efficient alternative to generate sentence representations in various languages.

A number of non-parameterized models have been proposed to derive sentence representations from pre-trained word embeddings (Rücklé et al., 2018; Yang et al., 2019; Kayal and Tsatsaronis, 2019). However, most of these models, including averaging, disregard structure information, which is an important aspect of any given language. Recently, Almarwani et al. (2019) proposed a structure-sensitive sentence encoder, which utilizes Discrete Cosine Transform (DCT) as an efficient alternative to averaging. The authors show that this approach is versatile and scalable because it relies only on word embeddings, which can be easily obtained from large unlabeled data. Hence, in principle, this approach can be adapted to different languages. Furthermore, having an efficient, ready-to-use language-independent sentence encoder can enable knowledge transfer between different languages in cross-lingual settings, empowering the development of efficient and performant NLP models for low-resource languages.

In this paper, we empirically investigate the generality of DCT representations across languages as both a single language model and a cross-lingual model in order to assess the effectiveness of DCT across different languages.

2 DCT as sentence Encoder

In signal processing domain DCT is used to decompose signal into component frequencies revealing dynamics that make up the signal and transitions within (Shu et al., 2017). Recently, DCT has been adopted as a way to compress textual information

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 419–426
August 1–6, 2021. ©2021 Association for Computational Linguistics

(Kayal and Tsatsaronis, 2019; Almarwani et al., 2019). A key observation in NLP is that word vectors obey laws of algebra King – Man + Woman = (approx.) Queen (Mikolov et al., 2013). Thus, given word embeddings, cast a sentence as a multi-dimensional signal over time, in which DCT is used to summarize the general feature patterns in word sequences and compress them into fixed-length vectors (Kayal and Tsatsaronis, 2019; Almarwani et al., 2019).

Mathematically, DCT is an invertible function that maps an input sequence of N real numbers to the coefficients of N orthogonal cosine basis functions of increasing frequencies (Ahmed et al., 1974). The DCT components are arranged in order of significance. The first coefficient (c[0]) represents the sum of the input sequence normalized by the square length, which is proportional to the average of the sequence (Ahmed et al., 1974). The lower-order coefficients represent lower signal frequencies which correspond to the overall patterns in the sequence. For example, DCT is used for compression by preserving only the coefficients with large magnitudes. These coefficients can be used to reconstruct the original sequence exactly using the inverse transform (Watson, 1994).

In NLP, Kayal and Tsatsaronis (2019) applied DCT at the word level to reduce the dimensionality of the embeddings size, while Almarwani et al. (2019) applied it along the sentence length as a way to compress each feature in the embedding space independently. In both implementations, the top coefficients are concatenated to generate the final representation for a sentence. As shown in (Almarwani et al., 2019), applying DCT along the features in the embeddings space renders representations that yield better results. Also, Zhu and de Melo (2020) noted that similar to vector averaging the DCT model proposed by (Almarwani et al., 2019) yields better overall performance compared to more complex encoders, thus, in this work, we adopt their implementation to extract sentence-level representations.

Specifically, given a sentence matrix $N \times d$, a sequence of DCT coefficients $c[0], c[1], ..., c[K]$ are calculated by applying the DCT type II along the d-dimensional word embeddings, where $c[K] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} v_n \cos \frac{\pi}{N}(n + \frac{1}{2})K$ (Shao and Johnson, 2008). Finally, a fixed-length sentence vector of size Kd is generated by concatenating the first K DCT coefficients, which we refer to as $c[0 : K]$.[1]

Task	Description
SentLen	Length prediction
WC	Word Content analysis
BShift	Word order analysis
TreeDepth	Tree depth prediction
Tense	Verb tense prediction
CoordInv	Coordination Inversion
SubjNum	Subject number prediction
ObjNum	Object number prediction
SOMO	Semantic odd man out

Table 1: Probing Tasks as described in (Conneau et al., 2018; Ravishankar et al., 2019).

3 Multi-lingual DCT Embeddings

3.1 Experimental Setups and Results

In our study, DCT is used to learn a separate encoder for each language from existing monolingual word embeddings. To evaluate DCT embeddings across different languages, we used the probing benchmark provided by Ravishankar et al. (2019), which includes a set of multi-lingual probing datasets.[2] The benchmark covers five languages: English, French, German, Spanish and Russian, derived from Wikipedia. The task set comprises 9 probing tasks, summarized in Table 1, that address varieties of linguistic properties including surface, syntactic, and semantic information (Conneau et al., 2018; Ravishankar et al., 2019). Ravishankar et al. (2019) used the datasets to evaluate different sentence encoders trained by mapping sentence representations to English. Unlike Ravishankar et al. (2019), we use the datasets to evaluate DCT embeddings for each language independently. As a baseline, in addition to the DCT embeddings, we use vector averaging to extract sentence representations from the pre-trained embeddings.

For model evaluations, we utilize the SentEval framework introduced in (Conneau and Kiela, 2018). In all experiments, we use a single-layer MLP on top of DCT sentence embeddings with the following parameters: kfold=10, batch_size=128, nhid=50, optim=adam, tenacity=5, epoch_size=4.

[1] Unlike (Almarwani et al., 2019), we note no further improvements with larger coefficients, thus, we only report the results of $1 \leq K \leq 4$.

[2] Refer to (Conneau et al., 2018) and (Ravishankar et al., 2019) for more details about the probing tasks.

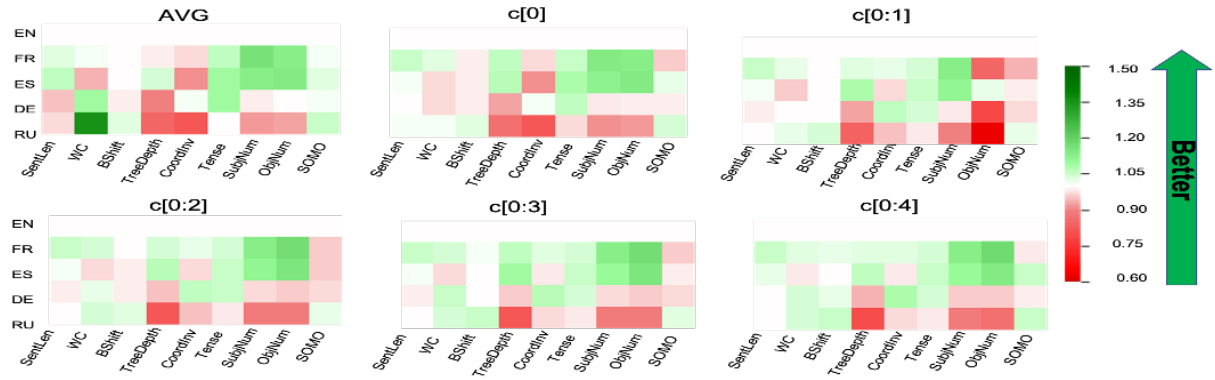


Figure 1: Results of the probing tasks comparing XX languages performance relative to English. White indicates a value of 1, demonstrating parity in performance with English. Red indicates better English performance while green indicates better XX Lang results.

For the word embeddings, we relied on the publicly available pre-trained FastText embeddings introduced in (Grave et al., 2018).[3]

Results: Figure 1 shows a heat-map reflecting the probing results of the different languages relative to English. Overall, French (FR) seems to be the closest to English (EN) followed by Spanish (ES) then German (DE) and then finally Russian (RU) across the various DCT coefficients. Higher coefficients reflect majority better performance across most tasks for FR, ES and DE. We see the most variation with worse results than English on the syntactic tasks of TreeDepth, CoordInv, Tense, SubjNum and ObjNum for RU. SOMO stands out for RU where it outperforms EN. The variation in Russian might be due to the nature of RU being a more complex language that is morphologically rich with flexible word order (Toldova et al., 2015).

In terms of the performance per number of DCT coefficients, we observe consistent performance gain across different languages that is similar to the English result trends. Specifically, for the surface level tasks, among the DCT models the $c[0]$ model significantly outperforms the AVG with an increase of $\sim$30 percentage points in all languages. The surface level tasks (SentLen and WC) show the most notable variance in performance, in which the highest results are obtained using the $c[0]$ model. However, the performance decreases in all languages when K is increased. On the other hand, for all languages, we observe a positive effect on the model's performance with larger K in both the syntactic and semantic tasks. The complete numerical results are presented in the Appendix in Table 5.

[3] Available at: https://fasttext.cc.

4 Cross-lingual Mapping based on DCT Encoding

4.1 Approach

Aldarmaki and Diab (2019)proposed sentence-level transformation approaches to learn context-aware representations for cross-lingual mappings. While the word-level cross-lingual transformations utilize an aligned dictionary of word embeddings to learn the mapping, the sentence-level transformations utilize a large dictionary of parallel sentence embeddings. Since sentences provide contexts that are useful for disambiguation for the individual word's specific meaning, sentence-level mapping yields a better cross-lingual representation compared to word-level mappings.

A simple model like sentence averaging can be used to learn transformations between two languages as shown in (Aldarmaki and Diab, 2019). However, the resulting vectors fail to capture structural information such as word order, which may result in poor cross-lingual alignment. Therefore, guided by the results shown in (Aldarmaki and Diab, 2019), we further utilize DCT to construct sentence representations for the sentence-level cross-lingual modeling.

4.2 Experiments Setups and Results

For model evaluation, we use the same cross-lingual evaluation framework introduced in (Aldarmaki and Diab, 2019). Intuitively, sentences tend to be clustered with their translations when their vectors exist in a well-aligned cross-lingual space. Thus, in this framework, cross-lingual mapping ap-

proaches are evaluated using sentence translation retrieval by calculating the accuracy of correct sentence retrieval. Formally, the cosine similarity is used to find the nearest neighbor for a given source sentence from the target side of the parallel corpus.

4.3 Evaluation Datasets and Results

To demonstrate the efficacy of cross-lingual mapping using the sentence-level representation generated by DCT models, similarly to Aldarmaki and Diab (2019), we used the WMT'13 data set that includes EN, ES and DE languages (Bojar et al., 2013). We further used five language pairs from the WMT'17 translation task to evaluate the effectiveness of DCT-based embeddings. Specifically, we used a sample of 1 million parallel sentences from WMT'13 common-crawl data; this subset is the same one used in (Aldarmaki and Diab, 2019).[4] To assess efficacy of the DCT models for the cross-lingual mapping, we reported the performances of the sentence translation retrieval task within the WMT'13 test set, which includes EN, ES, and DE as test languages (Bojar et al., 2013). Specifically, we first used the 1M parallel sentences for the alignment between source languages (ES and DE) to a target language (EN) independently. We evaluated the translation retrieval performance in all language directions, from source languages to English: ES-EN and DE-EN, as well as between the sources languages: ES-DE.

Similarly, we conduct a series of experiments on 5 different language pairs from the WMT'17 translation task, which includes DE, Latvian (LV), Finnish (FI), Czech (CS), and Russian (RU), each of which is associated with an English translation (Zhang et al., 2018).[5] For each language pair, we sampled 1M parallel sentences from their training corpus for the cross-lingual alignment between each source language and EN. Also, we used the test set available for each language pair to evaluate the translation retrieval performances.

In our experiments, we evaluate the translation retrieval performance in all language directions using three type of word embeddings: 1- a publicly available pre-trained word embeddings in which we show the performance of DCT against averaging, which we refer to hereafter as out-of-domain embeddings as shown in Table 2. 2- Also, we ran additional experiments in which we used a domain specific word embedding (that we trained on genre that is similar to the translation task) and 3-contextualized word embedding, which we refer to hereafter as in-domain embeddings as shown in Table 3.

Lang pair	AVG	$c[0]$	$c[0:1]$	$c[0:2]$	$c[0:3]$
Lang→EN					
ES→EN	65.67	64.87	71.26	**71.80**	70.13
DE→EN	51.80	50.30	57.23	**58.13**	56.57
RU→EN	45.22	52.75	61.91	**64.35**	63.33
CS→EN	41.87	42.50	52.89	54.99	**55.05**
FI→EN	40.46	42.00	47.57	**47.80**	46.16
LV→EN	21.26	40.13	51.42	56.37	**60.16**
EN→Lang					
EN→ES	69.97	69.50	73.73	**73.87**	71.73
EN→DE	67.50	66.23	**69.27**	68.70	65.83
EN→RU	38.09	44.29	54.73	59.51	**60.94**
EN→CS	39.73	40.40	50.99	54.00	**54.12**
EN→FI	39.34	42.52	51.67	**52.59**	51.74
EN→LV	15.83	33.55	47.08	53.22	**55.72**
Lang1→Lang2					
DE→ES	43.80	42.20	49.50	**51.20**	51.17
ES→DE	57.67	56.46	**60.53**	59.83	57.87

Table 2: Sentence translation retrieval accuracy based on out of domain pre-trained Fasttext embeddings. Arrows indicate the direction, with English (EN), Spanish (ES), German (DE), Russian (RU), Czech (CS), Finnish (FI) , Turkish (TR), and Latvian (LV).

Out-of-domain embeddings: For all language pairs, DCT-based models outperform AVG and c[0] models in the sentence translation retrieval task. In the direction $\rightarrow EN$, while the c[0:2] model achieve the highest accuracy for ES, DE, RU, and FI languages, the c[0:3] model achieved the highest accuracy for CS and LV languages. Specifically, the c[0:2] model yields increases of 5.59%-30% in the direction from source languages (ES, DE, RU, and FI) to English compared to the AVG model. Also, while the c[0:3] model yielded an increase of 13% gains over the baseline for CS, it provides the most notable increase of 38% for LV. For the opposite directions $EN \rightarrow source$, the DCT-based embeddings model also outperformed AVG and c[0] models. In particular, we observed accuracy gains of at least 3.81% points using more coefficients in DCT-based models compared to the AVG and c[0] models for all languages. A similar trend is observed in the zero-shot translation retrieval between the two non English languages (ES and DE), in which DCT-based models outperform the AVG and c[0] models.

[4]Evaluation scripts and WMT'13 dataset as described in (Aldarmaki and Diab, 2019) are available in https://github.com/h-aldarmaki/sent_translation_retrieval

[5]The pre-processed version of the WMT'17 dataset was used. For more information refer to (Zhang et al., 2018).

Lang pair	Embed	AVG	$c[0]$	$c[0:1]$	$c[0:2]$	$c[0:3]$
Lang→EN						
ES→EN	FT	82.97	82.40	**84.50**	83.97	82.90
	BERT	92.10	92.00	**93.23**	93.13	92.20
DE→EN	FT	79.33	78.73	**81.87**	80.20	77.93
	BERT	89.76	89.66	**91.83**	91.20	90.57
EN→Lang						
EN→ES	FT	82.33	82.07	**85.47**	84.60	83.17
	BERT	93.63	93.66	**94.10**	94.00	92.80
EN→DE	FT	74.73	74.50	**79.10**	78.70	76.90
	BERT	91.30	91.43	**91.90**	91.53	90.30
Lang1→Lang2						
DE→ES	FT	73.27	72.20	**77.43**	75.96	74.60
	BERT	87.80	87.57	90.23	**90.36**	88.96
ES→DE	FT	68.90	68.07	**73.97**	73.10	72.43
	BERT	87.70	87.70	**89.67**	89.50	88.53

Table 3: Accuracy using in-domain FastText (FT) and Contextualized mBERT embeddings. The best results for each row in **Bold** & for each direction in **gray**.

Model	**Average Accuracy**
FastText (dict) [ALD2019]	69.04
ELMo (word) [ALD2019]	82.23
FastText (word) [ALD2019]	74.00
FastText AVG (sent) [ALD2019]	76.92
ELMo AVG (sent) [ALD2019]	84.03
FastText $c[0]$ (sent)	76.33
FastText $c[0:1]$ (sent)	80.39
FastText $c[0:2]$ (sent)	79.42
FastText $c[0:3]$ (sent)	77.99
mBERT AVG (sent)	90.38
mBERT $c[0]$ (sent)	90.34
mBERT $c[0:1]$ (sent)	**91.83**
mBERT $c[0:2]$ (sent)	91.62
mBERT $c[0:3]$ (sent)	90.56

Table 4: The average accuracy of various models across all language retrieval directions as reported in (Aldarmaki and Diab, 2019), refer to as [ALD2019] in the table, along with the different DCT-based models in this work, in which (word) refers to word-level mapping, (sent) refers to sentence-level mapping, and (dict) refers to the baseline (using a static dictionary for mapping). **Bold** shows the best overall result.

In-domain embeddings: To ensure comparability to state-of-the-art results, we further utilized in-domain FastText embeddings as those used in (Aldarmaki and Diab, 2019) as well as contextualized-based word embeddings. For the in-domain FastText embeddings, the FastText (Bojanowski et al., 2017) is utilized to generate word embeddings from 1 Billion Word benchmark (Chelba et al., 2014) for English, and equivalent subsets of about 400 million tokens from WMT'13 (Bojar et al., 2013) news crawl data. For the contextualized-based embeddings, we utilized multilingual BERT (mBERT) introduced in (Devlin et al., 2019) as contextual word embeddings, in which representations from the last BERT layer are taken as word embeddings. As shown in Table 3, using in-domain word embeddings yields stronger results compared to the pre-trained embeddings we use in the previous experiments as illustrated in Table 2. On the other hand, we observe additional improvements using mBERT as word embeddings on all models. Furthermore, increasing K has positive effect on both embeddings, in which $c[0:1]$ demonstrate performance gains compared to other models in all language directions. This trend is clearly observed in the zero-shot performance between the non English languages.

Furthermore, as shown in Table 4, we obtained a state-of-the-art result using mBERT $c[0:1]$ with **91.83%** average accuracy across all translation directions compared to the 84.03% average accuracy of ELMo as reported in (Aldarmaki and Diab, 2019).

5 Conclusion

In this paper, we extended the application of DCT encoder to multi- and cross-lingual settings. Experimental results across different languages showed that similar to English using DCT outperform the vector averaging. We further presented a sentence-level-based approach for cross-lingual mapping without any additional training parameters. In this context, the DCT embedding is used to generate sentence representations, which are then used in the alignment process. Moreover, we have shown that incorporating structural information encoded in the lower-order coefficients yields significant performance gains compared to the AVG in sentence translation retrieval.

Acknowledgments

We thank Hanan Aldarmaki for providing us the in-domain FastText embeddings and for sharing many helpful insights. We would also like to thank 3 anonymous reviewers for their constructive feedback on this work.

References

Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. 1974. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93.

Hanan Aldarmaki and Mona Diab. 2019. Context-

aware cross-lingual mapping. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3906–3911, Minneapolis, Minnesota. Association for Computational Linguistics.

Nada Almarwani, Hanan Aldarmaki, and Mona Diab. 2019. Efficient sentence embedding using discrete cosine transform. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3672–3678, Hong Kong, China. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639. ISCA.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

Subhradeep Kayal and George Tsatsaronis. 2019. EigenSent: Spectral sentence embeddings using higher-order dynamic mode decomposition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4536–4546, Florence, Italy. Association for Computational Linguistics.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Vinit Ravishankar, Lilja Øvrelid, and Erik Velldal. 2019. Probing multilingual sentence representations with X-probe. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 156–168, Florence, Italy. Association for Computational Linguistics.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated power mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv:1803.01400*.

Xuancheng Shao and Steven G Johnson. 2008. Type-ii/iii dct/dst algorithms with reduced number of arithmetic operations. *Signal Processing*, 88(6):1553–1564.

Xiao Shu, Xiaolin Wu, and Bolin Liu. 2017. A study on quantization effects of dct based compression. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3500–3504. IEEE.

S Toldova, O Lyashevskaya, A Bonch-Osmolovskaya, and M Ionov. 2015. Evaluation for morphologically rich language: Russian nlp. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, page 300. The Steering Committee of The World Congress in Computer Science, Computer ....

Andrew B Watson. 1994. Image compression using the discrete cosine transform. *Mathematica journal*, 4(1):81.

Ziyi Yang, Chenguang Zhu, and Weizhu Chen. 2019. Parameter-free sentence embedding via orthogonal basis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 638–648, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Xunjie Zhu and Gerard de Melo. 2020. Sentence analogies: Linguistic regularities in sentence embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3389–3400, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Appendices

Table 5 shows the complete numerical results for the probing tasks on all languages.

	Language	AVG	c[0]	c[0:1]	c[0:2]	c[0:3]	c[0:4]
SentLen	EN	56.28	**89.03**	88.91	88.95	88.7	88.08
	ES	59.92	89.59	90.00	89.8	89.73	**90.05**
	FR	57.9	**93.72**	93.44	93.14	92.82	92.38
	DE	53.41	**88.81**	88.36	88.16	87.54	87.69
	RU	54.42	**89.66**	89.12	89.18	88.26	88.04
WC	EN	26.97	**66.69**	64.55	62.49	60.39	59.08
	ES	25.4	**64.80**	62.18	60.62	58.76	57.64
	FR	27.14	**68.60**	66.13	64.71	62.8	61.04
	DE	29.33	**64.99**	64.52	63.93	63.12	61.54
	RU	36.33	**67.50**	65.58	64.69	62.69	61.32
Bshift	EN	54.78	54.98	54.58	54.86	54.81	**55.58**
	ES	54.7	54.52	54.53	54.21	54.71	**55.77**
	FR	54.69	54.7	54.68	54.91	55.53	**56.50**
	DE	54.23	54.22	54.35	54.43	54.6	**56.46**
	RU	56.48	56.8	56.81	56.28	57.4	**58.51**
TreeDepth	EN	41.34	45.18	48.64	49.84	49.44	**50.47**
	ES	42.9	48.53	52.29	53.34	**53.87**	53.54
	FR	41.06	47.68	50.05	51.65	**52.27**	52.15
	DE	37.06	41.97	45.14	47.33	**47.55**	47.36
	RU	35.27	39.21	40.76	**41.02**	40.65	40.51
Tense	EN	86.49	89.23	91.83	92.17	**92.26**	92.21
	ES	94.52	95.97	**96.68**	96.67	96.62	96.53
	FR	91.96	94.06	95.7	95.96	**96.12**	95.99
	DE	94.13	94.71	95.82	**96.44**	96.28	95.92
	RU	86.07	86.39	90.28	**90.4**	90.16	90.38
CoordInv	EN	73.47	74.22	84.56	**87.20**	87.03	87.19
	ES	67.08	68.13	81.61	84.15	85.17	**85.77**
	FR	71.06	71.12	85.97	88.03	89.21	**89.61**
	DE	74.25	74.33	89.99	92.52	93.45	**94.09**
	RU	60.33	60.77	79.95	83.13	84.03	**84.34**
SubjNum	EN	76.46	77.41	80.49	81.68	81.76	**82.31**
	ES	86.4	86.68	89.34	90.42	90.12	**90.84**
	FR	88.48	88.62	91.05	92.23	92.72	**92.76**
	DE	75.94	75.78	78.79	78.9	79.25	**79.28**
	RU	70.47	70.44	72.31	72.81	73.12	**73.13**
ObjNum	EN	68.44	69.71	71.78	73.24	73.98	**74.93**
	ES	78.31	79.23	82.21	83.96	85.2	**85.7**
	FR	77.47	78.5	83.74	85.82	86.92	**88.1**
	DE	68.38	68.74	69.88	70.41	71.14	**71.90**
	RU	63.9	63.79	65.33	65.32	**65.54**	65.11
SOMO	EN	50.12	50.91	**51.72**	51.71	51.36	50.42
	ES	51.7	51.98	51.34	49.62	50.71	**53.07**
	FR	**50.7**	48.85	48.87	49.44	49.56	49.36
	DE	**50.57**	50.47	49.99	49.99	49.99	49.99
	RU	52.49	52.91	52.86	52.8	53.07	**53.13**

Table 5: DCT embeddings Performance per language compared to AVG. **EN**=English, **ES**=Spanish, **FR**=French, **DE**=German, and **RU**=Russian

AligNarr: Aligning Narratives on Movies

Paramita Mirza, **Mostafa Abouhamra**, and **Gerhard Weikum**
Max Planck Institute for Informatics
Saarbrücken, Germany
{paramita, mostafa, weikum}@mpi-inf.mpg.de

Abstract

High-quality alignment between movie scripts and plot summaries is an asset for learning to summarize stories and to generate dialogues. The alignment task is challenging as scripts and summaries substantially differ in details and abstraction levels as well as in linguistic register. This paper addresses the alignment problem by devising a fully unsupervised approach based on a global optimization model. Experimental results on ten movies show the viability of our method with 76% F1-score and its superiority over a previous baseline. We publish alignments for 914 movies to foster research in this new topic.

1 Introduction

Motivation and Problem. An important aspect of language understanding is the ability to produce a concise and fluent summary of stories, dialogues and other textual contents. Automatic text summarization is a long-standing topic in natural language processing (Nenkova and McKeown, 2012; Dey and Das, 2020), with numerous approaches for a variety of inputs, largely focusing on news articles and scholarly publications (e.g., See et al. (2017); Hardy et al. (2019); Lev et al. (2019)).

In this paper, our focus is on less explored *narrative* texts such as books and movie scripts. Our goal is to automatically align scenes from movie scripts with sentences from plot summaries. Such alignments support story browsing and explorative search over screenplays (e.g., find all love scenes), and can also be an asset towards improving summarization and text-generation models for dialogues and other narratives.

Figure 1 shows an example: a scene snippet from the movie script of *Shrek*, and its corresponding sentence from the plot description of the movie's Wikipedia article. Establishing this alignment is challenging for three reasons:

- *Input Length:* Movies have many scenes (often more than a hundred), with longer dialogues or multi-person conversations. Plot summaries, on the other hand, are much shorter (e.g., 700 words for *Shrek* on Wikipedia).
- *Disparate Registers:* Scripts and summaries have fundamentally different registers (i.e., language styles, vocabulary and structure). Scripts are dominated by direct speech in dialogues, whereas plot summaries consist of, often complex, descriptive sentences and may introduce abstractions (e.g. *"fell in love ..."* instead of giving details on dating, kissing etc.).
- *Disparate Granularities:* Scripts contain every detail of the screenplay, whereas summaries focus on salient points and can leave out less important sub-stories. Thus, the units in scripts–*scenes*–and the units in plot summaries–*sentences*–are difficult to match.

Narrative Alignment Task: Given a *script* S consisting of a sequence of m scenes $\{s_1, s_2, ..., s_m\}$ and a *summary* U of n sentences $\{u_1, u_2, ..., u_n\}$, the narrative alignment task is to find a mapping between S and U, where both sides can be partial (i.e., some scenes and some sentences are not mapped) and certain constraints are satisfied.

Prior Work and its Limitations. The task of aligning narratives across different registers, like script dialogues and plot summaries, has not received much attention before. Gorinski and Lapata (2015) proposed a graph-based summarization method for movie scripts, exploiting given alignments between script scenes and plot sentences, to select a chain of scenes representing a film's story. Their focus was on the generation of the textual summary, and the alignment itself was addressed merely by simple best-match heuristics based on Nelken and Shieber (2006). Nevertheless, as this work is the relatively closest to ours, it is treated as

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 427–433
August 1–6, 2021. ©2021 Association for Computational Linguistics

Script:

Suddenly the magic of the spell pulls Fiona away. She's lifted up into the air and she hovers there while the magic works around her. Suddenly Fiona's eyes open wide. She's consumed by the spell and then is slowly lowered to the ground.

SHREK: (going over to her) Fiona? Fiona. Are you all right?
FIONA: (standing up, she's still an ogre) Well, yes. But I don't understand. I'm supposed to be beautiful.
SHREK: But you ARE beautiful.

Summary:

Fiona is bathed in light as her curse is broken but is surprised that she is still an ogre, as she thought she would become beautiful, to which Shrek replies that she is beautiful

Figure 1: Snippet from *Shrek*'s script, and its summary sentence from *Shrek*'s Wikipedia article.

the baseline against which we evaluate our method. Tapaswi et al. (2015) used a graph-based method to compute an alignment between book chapters and video scenes using matching dialogues and characters as cues. As far as we know, our work is the first in-depth investigation of the narrative alignment task between movie scripts and plot summaries.

Approach and Contributions. We model the narrative alignment task as a global optimization over the possible pairs of scene-sentence mappings. To cope with disparate language registers, we devise embedding-based similarity measures. To cope with the length issue and different granularities, we design this for partial mappings where not all scenes and not all sentences need to be mapped. Typically, a notable subset of scenes is left out, but most sentences are aligned. To keep the alignments concise, we constrain the number of scenes that a sentence can be mapped to, and vice versa. Furthermore, we assume that script and summary both follow the chronology of events in the movie. This is modeled as a constraint for approximate order-preservation. All these considerations are cast into an Integer Linear Program (ILP).

The salient contributions of our work are:

- a fully unsupervised methodology using ILP for aligning two narratives, and
- an aligned corpus of movie scripts and plot summaries for 914 movies, which can serve as training data for text summarization and story generation tasks.

2 Approach

Our alignment method, AligNarr, has three steps: *(i) pre-processing*, which includes linking names found in both inputs, *(ii) building a similarity matrix* between the text units of the two narratives, and *(iii) constructing the alignment mapping* given the similarity matrix as input.

2.1 Pre-Processing

Given a movie script S and its summary U, we first segment them into corresponding units s_i and u_j, which are *scenes* and *sentences* respectively. An interior or exterior indicator 'INT.' or 'EXT.' is commonly used to mark a *scene heading*–separating different scenes–followed by a location or setting. A scene usually contains narrative descriptions as well as dialogue lines, as shown in Figure 1.

Linking Story Entities. We retrieve all phrases that are capitalized, as well as speaker names that start the dialogue lines in a given script, as *candidate names*, excluding the beginning of sentences. However, in movie scripts it is often the case that words are in all-capitals for emphasis, e.g., 'ARE' in Figure 1. Therefore, we first ran *Truecaser*[1] (Lita et al., 2003) to avoid having such words identified as candidate names.

For each pair of collected candidate names, we compute string similarity based on Levenshtein distance using *FuzzyWuzzy*[2]. Given the distance matrix between pairs of names, we then cluster the names using the DBSCAN algorithm (Ester et al., 1996) in order to have a cluster of names representing one *story entity*, e.g., E_{40}: {'Fiona', 'FIONA', 'Princess Fiona'}.

To resolve pronouns, we run *AllenNLP coreference resolution*[3], an end-to-end neural model (Lee et al., 2017) leveraging SpanBERT embeddings (Joshi et al., 2020). All occurrences of clustered names in the script and summary are then replaced with the corresponding entity identifier (e.g., E_{40}). Note that we only consider linking story entities appearing in the summary, since they represent a subset of story entities that are central to the story.

[1] github.com/nreimers/truecaser
[2] github.com/seatgeek/fuzzywuzzy
[3] demo.allennlp.org/coreference-resolution

2.2 Similarity Matrix

We investigate three methods to measure similarity between units of script S and summary U:

Document Relevance Score. After removing stop words and punctuation, we compute the relevance scores of script units $\{s_1, ..., s_m\}$ (as the document collection D), for a given summary unit u_j (as the query q), using a ranking function. In this work, we use BM25 (Robertson and Zaragoza, 2009), a TF-IDF-based ranking function.

Word Overlap Score. We consider the sum of intersecting story entities and words (excluding stop words) that are similar (e.g., 'married' in s_i and 'wedding' in u_j), weighted by their similarity scores. As the similarity score between two words, we take the cosine similarity of word2vec embeddings (Mikolov et al., 2013); words are considered to be similar if their cosine similarity is above 0.5.

Sentence Similarity Score. We first compute sentence embeddings for a given summary unit u_j and all sentences in a script unit s_i, using RoBERTa (Liu et al., 2019) in *Sentence-Transformers*[4] (Reimers and Gurevych, 2019) optimized for the task of Semantic Textual Similarity (stsb-roberta-large). Taken as the similarity score is the highest cosine similarity between u_j's embeddings and embeddings of sentences in s_i. For practical reasons, we only compute sentence similarity scores for pairs of script and summary units with non-zero word overlap scores.

2.3 Alignment Mapping

Given a similarity matrix between units of script $S = \{s_1, ..., s_m\}$ and summary $U = \{u_1, ..., u_n\}$, we devise an Integer Linear Programming (ILP) model to optimize the overall alignment mapping as follows:

Objective Function. We want to maximize the story coherence between S and U in terms of textual similarity between the units: $\max \sum_i \sum_j sim(s_i, u_j) \cdot X_{ij}$, where $sim(s_i, u_j)$ is a numeric feature indicating the similarity or relatedness of s_i and u_j resulting from the previous step, and X_{ij} is a decision variable: $X_{ij} = 1$ if s_i and u_j are aligned, 0 otherwise.

Constraints. We define the following constraints to make sure that the alignment mapping follows the linear constraint of both narratives:

- Each summary sentence can only be aligned with at most r scenes: $\sum_j X_{*j} \leq r$.
- Each summary sentence can only be aligned with a block of r consecutive scenes: $\sum_i \sum_j \sum_k X_{ij} + X_{kj} \leq 1$ if $k \geq i + r$ and $\sum_i \sum_j \sum_k X_{ij} + X_{kj} \leq 1$ if $k \leq i - r, i \geq r$.
- The next summary sentence can only be about the same or the next scenes: $\sum_i \sum_j \sum_k X_{ij} + X_{kj+1} \leq 1$ if $k < i, j < n - 1$.
- The previous summary sentence can only be about the same or the previous scenes: $\sum_i \sum_j \sum_k X_{ij} + X_{kj-1} \leq 1$ if $k > i, j > 0$.

Candidate Space Pruning. To speed up the ILP inference, we exclude pairs of script and summary units, s_i and u_j, which are unlikely to be aligned. We employ the following pruning conditions:

- Given a summary unit u_j, we only consider scenes that yield similarity scores above θ in the ranked list of scenes.
- Given the most similar scene s_{top} to a summary unit u_j, we only consider scenes s_i in which $sim(s_{top}, u_j) - sim(s_i, u_j) < \sigma$.
- The candidate pairs (s_i, u_j) are within the diagonal line boundaries as depicted in Figure 2, by considering only (i, j) pairs that satisfy $j < ni/m + \tau n$ and $i < mj/n + \tau m$ with hyper-parameter τ.

3 Experiments

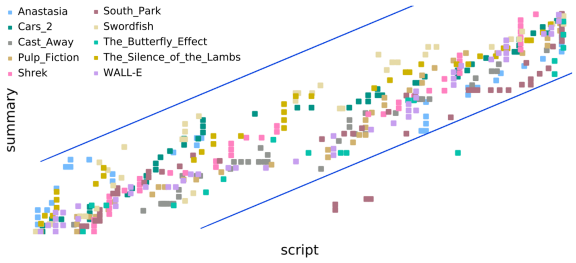


Figure 2: Ground truth alignment for ten movies.

Dataset. We used the ScriptBase corpus[5] (Gorinski and Lapata, 2015, 2018) that contains preprocessed scripts (with various automatic annotations and scene segmentation), along with the corresponding plot summaries taken from Wikipedia. Data statistics are given in Table 3. Two annotators manually created the alignment mappings for ten movies with varying script lengths, yielding inter-annotator agreement of 0.79 Fleiss' κ. The

[4] sbert.net/

[5] github.com/EdinburghNLP/scriptbase

	P	R	F1
Gorinski and Lapata (2015)	.520	.739	.482
AligNarr $_{\text{bm25}}$	.757	.719	.737
AligNarr $_{\text{bm25·w2v}}$	.789	.716	.746
AligNarr $_{\text{bm25·sts}}$	.789	**.734**	**.756**
AligNarr $_{\text{bm25·sts·w2v}}$	**.808**	.720	.754
AligNarr $_{\text{bm25·sts non-ILP}}$	.690	.717	.702

Table 1: AligNarr's performance against baseline.

movie	bm25·sts·w2v			bm25·sts·bert		
	P	R	F1	P	R	F1
Shrek	.85	.80	.82	.92↑	.90↑	.91↑
Pulp Fiction	.92	.86	.89	.89	.85	.87
Cars 2	.84	.72	.77	.87↑	.74↑	.79↑
The Silence of the Lambs	.86	.78	.81	.82	.78	.80
Anastasia	.87	.78	.82	.89↑	.79↑	.83↑
South Park: Bigger, Lo...	.83	.72	.76	.82	.71	.75
Wall-E	.92	.72	.79	.85	.74	.78
Swordfish	.75	.65	.68	.70	.61	.64
The Butterfly Effect	.63	.61	.62	.66↑	.61	.63↑
Cast Away	.61	.56	.58	.59	.56	.57
average	**.81**	**.72**	**.75**	**.80**	**.73↑**	**.76↑**

Table 2: AligNarr $_{\text{bm25·sts·w2v}}$ vs AligNarr $_{\text{bm25·sts·bert}}$.

ground-truth alignments (for which both annotators agree) are shown in Figure 2. These mappings confirm our intuition that a summary normally follows the corresponding script narration in a linear manner, with very few exceptions.

Hyper-Parameters. We defined r (in Section 2.3) as the average ratio of scenes to summary sentences $\lceil m/n \rceil$ based on ten movies, setting it to $r = 5$. θ was set to the 50^{th} percentile (i.e., the median). σ was set to the standard deviation of similarity scores for all scenes given the summary sentence u_j. Hyper-parameter τ, for pruning elements outside the diagonal line boundaries, was set to 0.3.

Baseline. Gorinski and Lapata (2015) used a classifier with sentence-level features (lemma overlap and word stem similarity) to compute sentence-to-sentence alignments. These aligned sentences were then used to identify aligned scene-sentence pairs forming the "gold chain" of scenes in this work (which focused more on the subsequent summarization task), in which a scene contains at least one sentence aligned with a summary sentence. They reported a precision of .53 at a recall rate of .82 for four movies. We re-ran their aligner (provided by the authors) on ten movies in our dataset.

4 Results and Discussion

We report macro-averaged precision (P), recall (R) and F1 results in Table 1. The best performing AligNarr variant, which runs ILP on the combination of document relevance and sentence similarity scores (AligNarr $_{\text{bm25·sts}}$) outperforms the baseline by a large margin on precision and F1-score.

Ablation Study. Document relevance scores alone (AligNarr $_{\text{bm25}}$) already yield very good performance with .737 F1-score averaged over ten movies. When combined with word overlap scores (AligNarr $_{\text{bm25·w2v}}$), the overall performance is further improved to .746 F1-score. Word overlap scoring using word embeddings is particularly useful when the summary uses different vocabulary, for example, using "...a *growing seedling*" for describing a scene with "...a small *plant* in its early stage of *growth*." Combining document relevance scores with sentence similarity scores (AligNarr $_{\text{bm25·sts}}$) results in the best performance with .756 F1 score. Adding word overlap scores on top of that (AligNarr $_{\text{bm25·sts·w2v}}$) yields higher precision of .808 but unfortunately at a lower recall rate of .720. Detailed comparisons and runtime are available in Appendix A and B.

We explored different strategies to combine the similarity matrices, and found element-wise matrix multiplication to perform the best.

Global vs. Local Alignments. To assess the benefit of using ILP, we devised an alignment algorithm focusing on finding the best scene alignment per summary sentence, that is, locally without using the ILP. Given a ranked list of scenes for a given summary sentence, we greedily pick scene-sentence pairs while observing the constraints on at most r consecutive scenes and the diagonal boundary for order-preservation. This local alignment algorithm results in .702 F1-score (AligNarr $_{\text{bm25·sts non-ILP}}$), showing the advantage of computing alignment mappings via global optimization.

Principal Limitation. The ILP constraints and diagonal line boundaries for candidate space pruning (presented in Section 2.3) are too restrictive to allow for 100% F1-score. Considering only candidate pairs that are within the diagonal line boundaries yields in reduced recall of .993, leading to F1-score of .997. If we also take into account all constraints employed by the ILP, recall is further reduced to .944, leading to F1-score of .969.

Contextual Embeddings. We also investigate the utility of contextual embeddings for computing word overlap scores. Specifically, we utilized a pretrained BERT model (bert-large-uncased) from Huggingface (https://huggingface.co/

movie	#scenes	#summary sentences	ratio	P	R	F1
Shrek	35	38	0.9	.89	.89	.89
Pulp Fiction	85	30	2.8	.89	.88	.88
Cars 2	113	36	3.1	.85	.75	.80
The Silence of the Lambs	136	28	4.9	.80	.79	.79
Anastasia	114	31	3.7	.80	.75	.77
South Park: Bigger, Lo...	120	41	2.9	.81	.72	.75
Wall-E	71	35	2.0	.84	.73	.77
Swordfish	*193*	*29*	*6.7*	*.76*	*.66*	*.70*
The Butterfly Effect	*182*	*17*	*10.7*	*.65*	*.61*	*.63*
Cast Away	*300*	*32*	*9.4*	*.60*	*.56*	*.58*
average	**135**	**32**	**4.7**	**.79**	**.73**	**.76**

Table 3: AligNarr $_{\text{bm25·sts}}$'s performance on ten movies.

transformers/model_doc/bert.html) to embed sentences from a given pair of script and summary units. We then retrieved individual vectors for each token (i.e., wordpiece) by summing together the outputs of BERT's last four layers.

For each token in the summary unit u_j, we look for similar tokens in the script unit s_i by computing cosine similarity of their embeddings, and take the highest one from each sentence as our intersecting tokens (only if their cosine similarity is above 0.7). Finally, BERT-based word overlap scores are the sum of overlapping tokens (excluding stop words) weighted by their cosine similarity.

Replacing word2vec embeddings with BERT embeddings (AligNarr $_{\text{bm25·sts·bert}}$ in Table 2) yields better performance for some movies like *Shrek*, *Cars 2* and *Anastasia*), which interestingly belong to the same genre (animation). The better performance may be attributed to the ability of BERT to better represent less common words (e.g., *ogre*) using contextual information. However, the overall performance is comparable with the performance of AligNarr $_{\text{bm25·sts·w2v}}$, which requires much less computing time (see Appendix B).

Movie Comparison. AligNarr's performance per movie is shown in Table 3. We observed a trend that the higher the ratio of scenes to summary sentences, the worse the alignment performance, particularly for three movies with ratio above r (average ratio, $r = 5$). This is potentially useful for estimating AligNarr's performance on other movies, which is negatively correlated to the compression rate of a given summary. The most difficult movie to align is *Cast Away*, where *(i)* there was only one active *story entity* throughout the narration, *(ii)* the summary is highly abstract (e.g., *"He also has regular conversations and arguments with Wilson."*), and *(iii)* the story plots and entity names do not fully match, possibly due to the outdated script version.

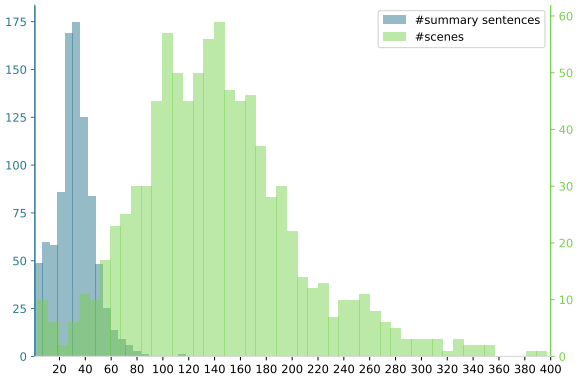


Figure 3: Histograms of number of scenes and summary sentences in ScriptBase movies.

Data and Code. We provide alignments by AligNarr for ten movies at d5demos.mpi-inf.mpg.de/alignarr/experiments; the same platform was used to manually annotate the alignment mappings. The code for producing the alignments is published at github.com/paramitamirza/AligNarr.

We applied the best performing AligNarr $_{\text{bm25·sts}}$ on the ScriptBase corpus[6] (Gorinski and Lapata, 2015, 2018), leveraging the XML version of movie scripts in *ScriptBase-J* and Wikipedia plot summaries from *ScriptBase-alpha*, totaling to 914 movies. Figure 3 shows the histograms of number of scenes and summary sentences in the corpus, with most summaries containing 20-40 sentences and most scripts consisting of around 100-180 scenes. The alignment mappings for those movies are made available for viewing and downloading at d5demos.mpi-inf.mpg.de/alignarr/scriptbase.

References

Monalisa Dey and Dipankar Das. 2020. A deep dive into supervised extractive and abstractive summarization from text. In *Data Visualization and Knowledge Engineering*, pages 109–132. Springer.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of KDD'96*, pages 226–231.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of NAACL-HLT'15*, pages 1066–1076.

[6] github.com/EdinburghNLP/scriptbase

Philip John Gorinski and Mirella Lapata. 2018. What's this movie about? a joint neural network architecture for movie content analysis. In *Proceedings of NAACL-HLT'18*, pages 1770–1781.

Hardy Hardy, Shashi Narayan, and Andreas Vlachos. 2019. HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings ACL'19*, pages 3381–3392.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of EMNLP'17*, pages 188–197.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. TalkSumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of ACL'19*, pages 2125–2131.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. tRuEcasIng. In *Proceedings of ACL'03*, pages 152–159.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*.

Rani Nelken and Stuart M. Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *In Proceedings of EACL'06*.

Ani Nenkova and Kathleen R. McKeown. 2012. A survey of text summarization techniques. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 43–76. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP'19*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL'17*, pages 1073–1083.

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. 2015. Book2movie: Aligning video scenes with book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1827–1835.

movie	AligNarr $_{bm25}$			AligNarr $_{bm25\cdot w2v}$			AligNarr $_{bm25\cdot sts}$			AligNarr $_{bm25\cdot sts\cdot w2v}$		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Shrek	.85	.84	.85	.85	.81	.83	.89↑	.89↑	.89↑	.85	.80	.82
Pulp Fiction	.85	.86	.86	.88↑	.85	.86	.89↑	.88↑	.88↑	.92↑	.86	.89↑
Cars 2	.86	.76	.80	.81	.71	.75	.85	.75	.80	.84	.72	.77
The Silence of the Lambs	.76	.73	.75	.85↑	.76↑	.80↑	.80↑	.79↑	.79↑	.86↑	.78	.81↑
Anastasia	.79	.75	.77	.87↑	.78↑	.82↑	.80↑	.75	.77	.87↑	.78↑	.82↑
South Park: Bigger, Lo...	.80	.74	.77	.79	.71	.74	.81↑	.72	.75	.83↑	.72	.76↑
Wall-E	.78	.71	.74	.86↑	.75↑	.79↑	.84↑	.73↑	.77↑	.92↑	.72	.79↑
Swordfish	.65	.62	.63	.71↑	.63↑	.66↑	.76↑	.66↑	.70↑	.75	.65	.68
The Butterfly Effect	.62	.61	.61	.62	.58	.60	.65↑	.61	.63↑	.63	.61	.62
Cast Away	.61	.57	.59	.65↑	.58↑	.61↑	.60	.56	.58	.61↑	.56	.58
average	**.76**	**.72**	**.74**	**.79↑**	**.72**	**.75↑**	**.79↑**	**.73↑**	**.76↑**	**.81↑**	**.72**	**.75**

Table 4: AligNarr's performance on ten movies (ablation study).

movie	text pre-processing	ILP (bm25·sts)	computing similarity matrix bm25	w2v	sts	bert
Shrek	5.5	4.1	0.02	69.7	131.3	1972.5
Pulp Fiction	20.8	12.3	0.03	222.0	172.0	2576.2
Cars 2	56.5	36.1	0.04	168.9	210.2	3169.8
The Silence of the Lambs	29.5	29.4	0.04	329.1	199.9	2439.2
Anastasia	21.6	28.6	0.03	147.6	147.8	1919.6
South Park: Bigger, Lo...	62.2	1350.4	0.04	165.4	251.9	3603.4
Wall-E	6.2	11.6	0.02	172.5	169.3	2771.6
Swordfish	17.0	457.4	0.04	143.4	152.6	1893.6
The Butterfly Effect	8.1	84.2	0.05	233.9	130.0	1297.2
Cast Away	8.3	329.4	0.04	237.4	294.8	3069.5
average	**23.6**	**234.4**	**0.04**	**189.0**	**186.0**	**2471.2**
computing infrastructure	4x Intel(R) Xeon(R) Gold 6136 # of cores: 48 # of threads: 96 Memory: 1.5TB		1x AMD EPYC 7502P # of cores: 32 # of threads: 64 Memory: 1TB GPU: 4x NVIDIA Quadro RTX 8000, 48 GB GDDR6			

Table 5: AligNarr's runtime (in seconds).

A Detailed Ablation Study

We report in Table 4 the ablation study on AligNarr's performance using different similarity matrices on ten movies. In general, leveraging word-based (w2v) and sentence-based (sts) semantic similarity scores via embeddings, in addition to document relevance scores (bm25), results in significantly higher precision for some movies, while recall remains more or less stable.

B AligNarr's Runtime

In Table 5 we detail the average runtime of the best performing AligNarr $_{bm25\cdot sts}$, along with the computing infrastructure used.

Note that to compute sentence similarity scores (sts) we need word overlap scores via word2vec embeddings (w2v) to filter out scene-sentence pairs that are unlikely to be similar, in order to speed up the runtime. Computing word overlap scores using BERT embeddings (bert) requires almost 13 times the time of computing the scores with word2vec embeddings.

An Exploratory Analysis of Multilingual Word-Level Quality Estimation with Cross-Lingual Transformers

Tharindu Ranasinghe[♢], **Constantin Orăsan**[♡] **and Ruslan Mitkov**[♢]
[♢]Research Group in Computational Linguistics, University of Wolverhampton, UK
[♡]Centre for Translation Studies, University of Surrey, UK
{t.d.ranasinghehettiarachchige, r.mitkov}@wlv.ac.uk
c.orasan@surrey.ac.uk

Abstract

Most studies on word-level Quality Estimation (QE) of machine translation focus on language-specific models. The obvious disadvantages of these approaches are the need for labelled data for each language pair and the high cost required to maintain several language-specific models. To overcome these problems, we explore different approaches to multilingual, word-level QE. We show that multilingual QE models perform on par with the current language-specific models. In the cases of zero-shot and few-shot QE, we demonstrate that it is possible to accurately predict word-level quality for any given new language pair from models trained on other language pairs. Our findings suggest that the word-level QE models based on powerful pre-trained transformers that we propose in this paper generalise well across languages, making them more useful in real-world scenarios.

1 Introduction

Quality Estimation (QE) is the task of assessing the quality of a translation without having access to a reference translation (Specia et al., 2009). Translation quality can be estimated at different levels of granularity: word, sentence and document level (Ive et al., 2018). So far the most popular task has been sentence-level QE (Specia et al., 2020), in which QE models provide a score for each pair of source and target sentences. A more challenging task, which is currently receiving a lot of attention from the research community, is word-level quality estimation. This task provides more fine-grained information about the quality of a translation, indicating which words from the source have been incorrectly translated in the target, and whether the words inserted between these words are correct (good vs bad gaps). This information can be useful for post-editors by indicating the parts of a sentence on which they have to focus more.

Word-level QE is generally framed as a supervised ML problem (Kepler et al., 2019; Lee, 2020) trained on data in which the correctness of translation is labelled at word-level (i.e. good, bad, gap). The training data publicly available to build word-level QE models is limited to very few language pairs, which makes it difficult to build QE models for many languages. From an application perspective, even for the languages with resources, it is difficult to maintain separate QE models for each language since the state-of-the-art neural QE models are large in size (Ranasinghe et al., 2020b).

In our paper, we address this problem by developing multilingual word-level QE models which perform competitively in different domains, MT types and language pairs. In addition, for the first time, we propose word-level QE as a zero-shot cross-lingual transfer task, enabling new avenues of research in which multilingual models can be trained once and then serve a multitude of languages and domains. The main contributions of this paper are the following:

i We introduce a simple architecture to perform word-level quality estimation that predicts the quality of the words in the source sentence, target sentence and the gaps in the target sentence.

ii We explore multilingual, word-level quality estimation with the proposed architecture. We show that multilingual models are competitive with bilingual models.

iii We inspect few-shot and zero-shot word-level quality estimation with the bilingual and multilingual models. We report how the source-target direction, domain and MT type affect the predictions for a new language pair.

iv We release the code and the pre-trained models as part of an open-source framework[1].

[1]Documentation is available on http://tharindu.co.uk/TransQuest/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 434–440
August 1–6, 2021. ©2021 Association for Computational Linguistics

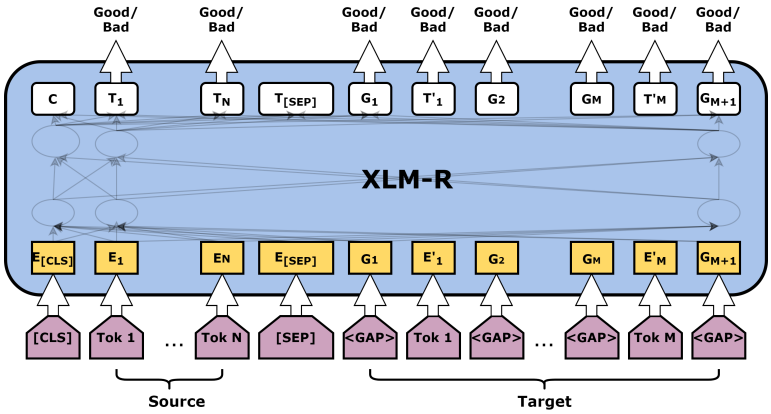


Figure 1: Model Architecture

2 Related Work

Quality Estimation Early approaches in word-level QE were based on features fed into a traditional machine learning algorithm. Systems like QuEst++ (Specia et al., 2015) and MARMOT (Logacheva et al., 2016) were based on features used with Conditional Random Fields to perform word-level QE. With deep learning models becoming popular, the next generation of word-level QE algorithms were based on bilingual word embeddings fed into deep neural networks. Such approaches can be found in OpenKiwi (Kepler et al., 2019). However, the current state of the art in word-level QE is based on transformers like BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) where a simple linear layer is added on top of the transformer model to obtain the predictions (Lee, 2020). All of these approaches consider quality estimation as a language-specific task and build a different model for each language pair. This approach has many drawbacks in real-world applications, some of which are discussed in Section 1.

Multilinguality Multilinguality allows training a single model to perform a task from and/or to multiple languages. Even though this has been applied to many tasks (Ranasinghe and Zampieri, 2020, 2021) including NMT (Nguyen and Chiang, 2017; Aharoni et al., 2019), multilingual approaches have been rarely used in QE (Sun et al., 2020). Shah and Specia (2016) explore QE models for more than one language where they use multitask learning with annotators or languages as multiple tasks. They show that multilingual models led to marginal improvements over bilingual ones with a traditional black-box, feature-based approach. In a recent study, Ranasinghe et al. (2020b) show that multilingual QE models based on transformers trained on high-resource languages can be used for zero-shot, sentence-level QE in low-resource languages. In a similar architecture, but with multi-task learning, Sun et al. (2020) report that multilingual QE models outperform bilingual models, particularly in less balanced quality label distributions and low-resource settings. However, these two papers are focused on sentence-level QE and to the best of our knowledge, no prior work has been done on multilingual, word-level QE models.

3 Architecture

Our architecture relies on the XLM-R transformer model (Conneau et al., 2020) to derive the representations of the input sentences. XLM-R has been trained on a large-scale multilingual dataset in 104 languages, totalling 2.5TB, extracted from the CommonCrawl datasets. It is trained using only RoBERTa's (Liu et al., 2019) masked language modelling (MLM) objective. XML-R was used by the winning systems in the recent WMT 2020 shared task on sentence-level QE (Ranasinghe et al., 2020a; Lee, 2020; Fomicheva et al., 2020). This motivated us to use a similar approach for word-level QE.

Our architecture adds a new token to the XLM-R tokeniser called <GAP> which is inserted between the words in the target. We then concatenate the source and the target with a [SEP] token and we feed them into XLM-R. A simple linear layer is added on top of word and <GAP> embeddings to predict whether it is "Good" or "Bad" as shown in Figure 1. The training configurations and the system specifications are presented in the supplementary material.

Language Pair	Source	MT System	Competition	Train Size
De-En	Pharmaceutical	Phrase-based SMT	WMT 2018	25,963
En-Cs	IT	Phrase-based SMT	WMT 2018	40,254
En-De	Wiki	fairseq-based NMT	WMT 2020	7,000
En-De	IT	fairseq-based NMT	WMT 2019	13,442
En-De	IT	Phrase-based SMT	WMT 2018	26,273
En-Ru	IT	Online NMT	WMT 2019	15,089
En-Lv	Pharmaceutical	Attention-based NMT	WMT 2018	12,936
En-Lv	Pharmaceutical	Phrase-based SMT	WMT 2018	11,251
En-Zh	Wiki	fairseq-based NMT	WMT 2020	7,000

Table 1: Information about the language pairs used to predict word-level quality. The **Language Pair** column lists the language pairs we used in ISO 639-1 codes. **Source** stands for the domain of the sentence and **MT System** is the Machine Translation system used to translate the sentences. **Competition** refers to the quality estimation competition in which the data was released and the last column indicates the number of instances the train dataset has for each language pair respectively.

4 Experimental Setup

4.1 QE Dataset

We used several language pairs for which word-level QE annotations were available: English-Chinese (En-Zh), English-Czech (En-Cs), English-German (En-De), English-Russian (En-Ru), English-Latvian (En-Lv) and German-English (De-En). The texts are from a variety of domains and the translations were produced using both neural and statistical machine translation systems. More details about these datasets can be found in Table 1 and in (Specia et al., 2018; Fonseca et al., 2019; Specia et al., 2020).

4.2 Evaluation Criteria

For evaluation, we used the approach proposed in the WMT shared tasks in which the classification performance is calculated using the multiplication of F1-scores for the 'OK' and 'BAD' classes against the true labels independently: words in the target ('OK' for correct words, 'BAD' for incorrect words), gaps in the target ('OK' for genuine gaps, 'BAD' for gaps indicating missing words) and source words ('BAD' for words that lead to errors in the target, 'OK' for other words) (Specia et al., 2018). In recent WMT shared tasks, the most popular category was predicting quality for words in the target. Therefore, in Section 5 we only report the F1-score for words in the target. Other results are presented in the supplementary material. Prior to WMT 2019, organisers provided separate scores for gaps and words in the target, while after WMT 2019 they produce a single result for target gaps and words. We follow this latter approach.

5 Results

The values displayed diagonally across section I of Table 2 show the results for supervised, bilingual, word-level QE models where the model was trained on the training set of a particular language pair and tested on the test set of the same language pair. As can be seen in section V, the architecture outperforms the baselines in all the language pairs and also outperforms the majority of the best systems from previous competitions. In addition to the target word F1-score, our architecture outperforms the baselines and best systems in target gaps F1-score and source words F1-score too as shown in Tables 5 and 6. In the following sections we explore its behaviour in different multilingual settings.

5.1 Multilingual QE

We combined instances from all the language pairs and built a single word-level QE model. Our results, displayed in section II ("All") of Table 2, show that multilingual models perform on par with bilingual models or even better for some language pairs. We also investigate whether combining language pairs that share either the same domain or MT type can be more beneficial, since it is possible that the learning process is better when language pairs share certain characteristics. However as shown in sections III and IV of Table 2, for the majority of the language pairs, specialised multilingual models built on certain domains or MT types do not perform better than multilingual models which contain all the data.

	Train Language(s)	IT				Pharmaceutical			Wiki	
		En-Cs SMT	En-De NMT	En-De SMT	En-Ru NMT	De-En SMT	En-LV NMT	En-Lv SMT	En-De NMT	En-Zh NMT
I	En-Cs SMT	0.6081	(-0.09)	(-0.07)	(-0.09)	(-0.15)	(-0.02)	(-0.01)	(-0.10)	(-0.11)
	En-De NMT	(-0.17)	0.4421	(-0.06)	(-0.02)	(-0.18)	(-0.01)	(-0.02)	(-0.01)	(-0.08)
	En-De SMT	(-0.01)	(-0.05)	0.6348	(-0.67)	(-0.14)	(-0.06)	(-0.04)	(-0.06)	(-0.09)
	En-Ru NMT	(-0.14)	(-0.08)	(-0.16)	**0.5592**	(-0.12)	(-0.01)	(-0.03)	(-0.09)	(-0.08)
	De-En SMT	(-0.43)	(-0.23)	(-0.33)	(-0.31)	**0.6485**	(-0.29)	(-0.32)	(-0.25)	(-0.28)
	En-LV NMT	(-0.12)	(-0.09)	(-0.14)	(-0.03)	(-0.12)	0.5868	(-0.01)	(0.09)	(-0.08)
	En-Lv SMT	(-0.04)	(-0.16)	(-0.10)	(-0.09)	(-0.16)	(-0.01)	0.5939	(-0.15)	(-0.14)
	En-De NMT	(-0.11)	(-0.01)	(-0.08)	(-0.02)	(-0.14)	(-0.02)	(-0.04)	0.5013	(-0.06)
	En-Zh NMT	(-0.19)	(-0.08)	(-0.17)	(-0.03)	(-0.16)	(-0.03)	(-0.06)	(-0.07)	0.5402
II	All	**0.6112**	**0.4523**	**0.6583**	0.5558	0.6221	**0.5991**	**0.5980**	0.5101	0.5229
	All-1	(-0.01)	(-0.01)	(-0.05)	(-0.02)	(-0.12)	(-0.01)	(-0.01)	(-0.01)	(-0.05)
III	Domain	0.6095	0.4467	0.6421	0.5560	0.6331	0.5892	0.5951	0.5021	0.5210
IV	SMT/NMT	0.6092	0.4461	0.6410	0.5421	0.6320	0.5885	0.5934	0.5010	0.5205
V	Baseline-Marmot	0.4449	0.1812	0.3630	NR	0.4373	0.4208	0.3445	NR	NR
	Baseline-OpenKiwi	NR	NR	NR	0.2412	NR	NR	NR	0.4111	0.5583
	Best system	0.4449	0.4361	0.6246	0.4780	0.6012	0.4293	0.3618	**0.6186**	**0.6415**

Table 2: Target F1-Multi between the algorithm predictions and human annotations. Best results for each language by any method are marked in bold. Sections I, II and III indicate the different evaluation settings. Section IV shows the results of the state-of-the-art methods and the best system submitted for the language pair in that competition. **NR** implies that a particular result was *not reported* by the organisers. Zero-shot results are coloured in grey and the value shows the difference between the best result in that section for that language pair and itself.

5.2 Zero-shot QE

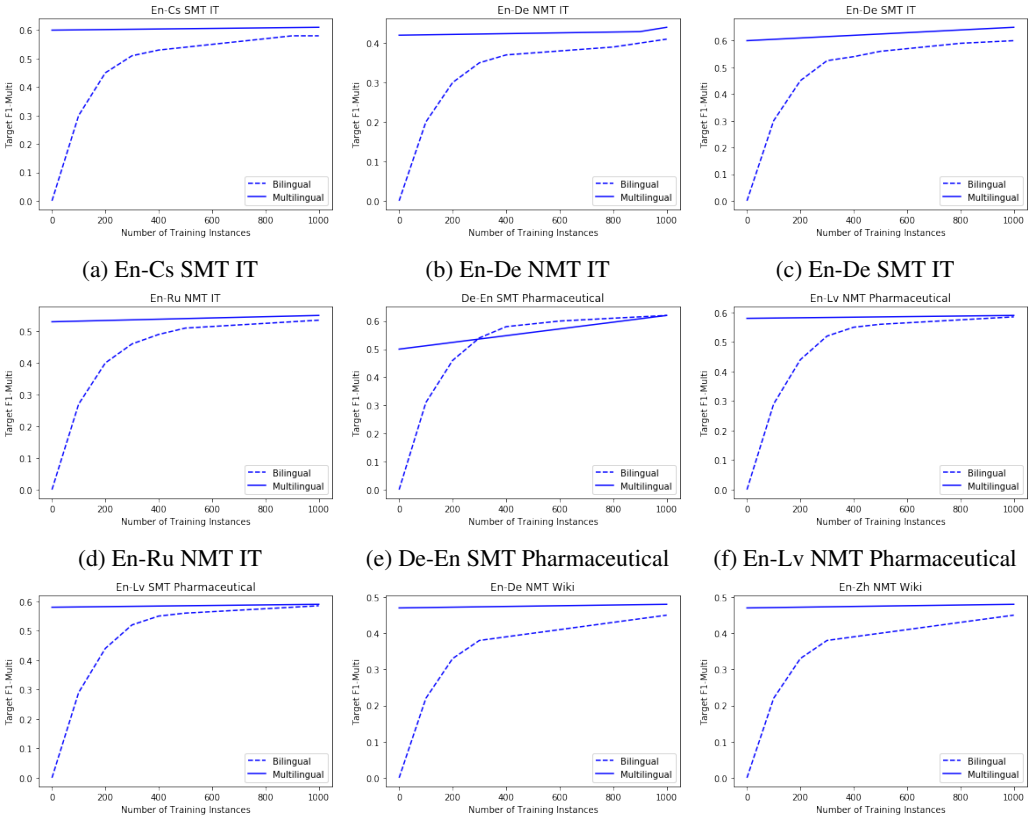
To test whether a QE model trained on a particular language pair can be generalised to other language pairs, different domains and MT types, we performed zero-shot quality estimation. We used the QE model trained on a particular language pair and evaluated it on the test sets of the other language pairs. Non-diagonal values of section I in Table 2 show how each QE model performed on other language pairs. For better visualisation, the non-diagonal values of section I of Table 2 show by how much the score changes when the zero-shot QE model is used instead of the bilingual QE model. As can be seen, the scores decrease, but this decrease is negligible and is to be expected. For most pairs, the QE model that did not see any training instances of that particular language pair outperforms the baselines that were trained extensively on that particular language pair. Further analysing the results, we can see that zero-shot QE performs better when the language pair shares some properties such as domain, MT type or language direction. For example, En-De SMT $\Rightarrow$ En-Cs SMT is better than En-De NMT $\Rightarrow$ En-Cs SMT and En-De SMT $\Rightarrow$ En-De NMT is better than En-Cs SMT $\Rightarrow$ En-De NMT.

We also experimented with zero-shot QE with multilingual QE models. We trained the QE model in all the pairs except one and performed prediction on the test set of the language pair left out. In section II ("All-1"), we show its difference to the multilingual QE model. This also provides competitive results for the majority of the languages, proving it is possible to train a single multilingual QE model and extend it to a multitude of languages and domains. This approach provides better results than performing transfer learning from a bilingual model.

One limitation of the zero-shot QE is its inability to perform when the language direction changes. In the scenario where we performed zero-shot learning from De-En to other language pairs, results degraded considerably from the bilingual result. Similarly, the performance is rather poor when we test on De-En for the multilingual zero-shot experiment as the direction of all the other pairs used for training is different. This is in line with results reported by Ranasinghe et al. (2020b) for sentence level.

5.3 Few-shot QE

We also evaluated how the QE models behave with a limited number of training instances. For each language pair, we initiated the weights of the bilingual model with those of the relevant All-1 QE and trained it on 100, 200, 300 and up to 1000 training instances. We compared the results with those obtained having trained the QE model from scratch for that language pair. The results in Figure 2 show



(a) En-Cs SMT IT (b) En-De NMT IT (c) En-De SMT IT

(d) En-Ru NMT IT (e) De-En SMT Pharmaceutical (f) En-Lv NMT Pharmaceutical

(g) En-Lv SMT Pharmaceutical (h) En-De NMT Wiki (i) En-Zh NMT Wiki

Figure 2: Target F1-Multi scores with Few-shot learning

that All-1 or the multilingual model performs well above the QE model trained from scratch (Bilingual) when there is a limited number of training instances available. Even for the De-En language pair, for which we had comparatively poor zero-shot results, the multilingual model provided better results with a few training instances. It seems that having the model weights already fine-tuned in the multilingual model provides an additional boost to the training process which is advantageous in a few-shot scenario.

6 Conclusions

In this paper, we explored multilingual, word-level QE with transformers. We introduced a new architecture based on transformers to perform word-level QE. The implementation of the architecture, which is based on Hugging Face (Wolf et al., 2020), has been integrated into the TransQuest framework (Ranasinghe et al., 2020b) which won the WMT 2020 QE task (Specia et al., 2020) on sentence-level direct assessment (Ranasinghe et al., 2020a)[2].

In our experiments, we observed that multilingual QE models deliver excellent results on the language pairs they were trained on. In addition, the multilingual QE models perform well in the majority of the zero-shot scenarios where the multilingual QE model is tested on an unseen language pair. Furthermore, multilingual models perform very well with few-shot learning on an unseen language pair when compared to training from scratch for that language pair, proving that multilingual QE models are effective even with a limited number of training instances. While we centered our analysis around the F1-score of the target words, these findings are consistent with the F1-score of the target gaps and the F1-score of the source words too. This suggests that we can train a single multilingual QE model on as many languages as possible and apply it on other language pairs as well. These findings can be beneficial to perform QE in low-resource languages for which the training data is scarce and when maintaining several QE models for different language pairs is arduous.

[2]TransQuest is available on GitHub https://github.com/tharindudr/TransQuest

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020. BERGAMOT-LATTE submissions for the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017, Online. Association for Computational Linguistics.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepQuest: A framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2016. MARMOT: A toolkit for translation quality estimation at the word level. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3671–3674, Portorož, Slovenia. European Language Resources Association (ELRA).

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5838–5844, Online. Association for Computational Linguistics.

Tharindu Ranasinghe and Marcos Zampieri. 2021. MUDES: Multilingual detection of offensive spans. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 144–152, Online. Association for Computational Linguistics.

Kashif Shah and Lucia Specia. 2016. Large-scale multitask learning for machine translation quality estimation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 558–567, San Diego, California. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán,

and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with QuEst++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Shuo Sun, Marina Fomicheva, Frédéric Blain, Vishrav Chaudhary, Ahmed El-Kishky, Adithya Renduchintala, Francisco Guzmán, and Lucia Specia. 2020. An exploratory study on multilingual quality estimation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 366–377, Suzhou, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Exploration and Exploitation: Two Ways to Improve Chinese Spelling Correction Models

Chong Li[†], **Cenyuan Zhang**[†], **Xiaoqing Zheng**[*], **Xuanjing Huang**
School of Computer Science, Fudan University, Shanghai, China
Shanghai Key Laboratory of Intelligent Information Processing
{chongli17, cenyuanzhang17, zhengxq, xjhuang}@fudan.edu.cn

Abstract

A sequence-to-sequence learning with neural networks has empirically proven to be an effective framework for Chinese Spelling Correction (CSC), which takes a sentence with some spelling errors as input and outputs the corrected one. However, CSC models may fail to correct spelling errors covered by the confusion sets, and also will encounter unseen ones. We propose a method, which continually identifies the weak spots of a model to generate more valuable training instances, and apply a task-specific pre-training strategy to enhance the model. The generated adversarial examples are gradually added to the training set. Experimental results show that such an adversarial training method combined with the pre-training strategy can improve both the generalization and robustness of multiple CSC models across three different datasets, achieving state-of-the-art performance for CSC task.[1]

1 Introduction

Chinese Spelling Correction (CSC) aims to detect and correct spelling mistakes in Chinese texts. Many Chinese characters are visually or phonologically similar, while their semantic meaning may differ greatly. Spelling errors are usually caused by careless writing, automatic speech recognition, and optical character recognition systems. The CSC task has received steady attention over the past two decades (Chang, 1995; Xin et al., 2014; Wang et al., 2018; Hong et al., 2019). Unlike English, Chinese texts are written without using whitespace to delimit words, and it is hard to identify whether and which characters are misspelled without the information of word boundaries. The context information should be taken into account to reconstruct the word boundaries when correcting spelling mistakes, which makes CSC a long-standing challenge for Chinese NLP community.

Many early CSC systems follow the same recipe with minor variations, adopting a three-step strategy: detect the positions of spelling errors; generate candidate characters for these positions; and select a most appropriate one from the candidates to replace the misspelling (Yeh et al., 2013; Yu and Li, 2014; Zhang et al., 2015; Wang et al., 2019). Recently, a sequence-to-sequence (seq2seq) learning framework with neural networks has empirically proven to be effective for CSC, which transforms a sentence with errors to the corrected one (Zhang et al., 2020; Cheng et al., 2020b).

However, even if training a CSC model with the seq2seq framework normally requires a huge amount of high-quality training data, it is still unreasonable to assume that all possible spelling errors have been covered by the *confusion* sets (i.e. a set of characters and their visually or phonologically similar characters which can be potentially confused) extracted from the training samples. New spelling errors occur everyday. A good CSC model should be able to exploit what it has already seen in the training instances in order to achieve reasonable performance on easy spelling mistakes, but it can also explore in order to generalize well to possible unseen misspellings.

In this study, we would like to pursue both the *exploration* (unknown misspellings) and *exploitation* (the spelling errors covered by the confusion sets) when training the CSC models. To encourage a model to explore unknown cases, we propose a character substitution-based method to pre-train the model. The training data generator chooses about 25% of the character positions at random for prediction. If a character is chosen, we replace it with the character randomly selected from its confusion set (90% of the time) or a random character (10%

[†]These authors contributed equally to this work.
[*]Corresponding Author
[1]The source codes are available at https://github.com/FDChongli/TwoWaysToImproveCSC.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 441–446
August 1–6, 2021. ©2021 Association for Computational Linguistics

of the time). Then, the model is asked to predict the original character.

Because of the combination of spelling errors and various contexts in which they occur, even though the confusion sets are given and fixed, models may still fail to correct characters that are replaced by any character from its confusion set. To better exploit what the models has experienced during the training phase, we generate more valuable training data via adversarial attack (i.e. tricking models to make false prediction by adding imperceptible perturbation to the input (Szegedy et al., 2014)), targeting the weak spots of the models, which can improve both the quality of training data for fine-tuning the CSC models and their robustness against adversarial attacks. Inspired by adversarial attack and defense in NLP (Jia and Liang, 2017; Zhao et al., 2018; Cheng et al., 2020a; Wang and Zheng, 2020), we propose a simple but efficient method for adversarial example generation: we first identify the most vulnerable characters with the lowest generation probabilities estimated by a pre-trained model, and replace them with characters from their confusion sets to create the adversarial examples.

Once the adversarial examples are obtained, they can be merged with the original clean data to train the CSC models. The examples generated by our method are more valuable than those already existed in the training set because they are generated towards to the weak spots of the current models. Through extensive experimentation, we show that such adversarial examples can improve both generalization and robustness of CSC models. If a model pre-trained with our proposed character substitution-based method is further fine-tuned by adversarial training, its robustness can be improved about 3.9% while without suffering too much loss (less than 1.1%) on the clean data.

2 Method

2.1 Problem Definition

Chinese Spelling Correction aims to identify incorrectly used characters in Chinese texts and giving its correct version. Given an input Chinese sentence $X = \{x_1, ..., x_n\}$ consisting of n characters, which may contain some spelling errors, the model takes X as input and outputs an output sentence $Y = \{y_1, ..., y_n\}$, where all the incorrect characters are expected to be corrected. This task can be formulated as a conditional generation problem by modeling and maximizing the conditional probability of $P(Y|X)$.

2.2 Base Models

We use vanilla BERT (Devlin et al., 2019) and two recently proposed BERT-based models (Cheng et al., 2020b; Zhang et al., 2020) as our base models. When applying BERT to the CSC task, the input is a sentence with spelling errors, and the output representations are fed into an output layer to predict target tokens. We tie the input and output embedding layer, and all the parameters are fine-tuned using task-specific corpora. Soft-Masked BERT (Zhang et al., 2020) uses a Bi-GRU network to detect errors, and applies a BERT-based network to correct errors. SpellGCN (Cheng et al., 2020b) utilizes visual and phonological similarity knowledge through a specialized graph convolutional network and substitutes parameters of the output layer of BERT with the final output of it.

These models achieved state-of-the-art or close to state-of-the-art performance on the CSC task. However, we found that their performance and robustness could be further improved through pre-training and adversarial training, which help models explore unseen spelling errors and exploit weak points of themselves.

2.3 Pre-training Method

We collected unlabeled sentences from Wikipedia and Weibo corpora (Shang et al., 2015), covering both formal and informal Chinese texts. Training example pairs are generated by substituting characters in clean sentences, and models are trained to predict the original character. According to Chen et al. (2011), a sentence contains no more than two spelling errors on average, so we select and replace 25% characters in a sentence. The chosen Chinese character will be substituted by a character randomly selected from its confusion set (90% of the time) or a random Chinese character (10% of the time). The latter helps models to explore unknown misspellings not covered by the confusion sets.

2.4 Adversarial Example Generation and Adversarial Training

To efficiently identify and alleviate the weak spots of trained CSC models, we designed an adversarial attack algorithm for CSC tasks, which replaces the tokens in a sentence with spelling mistakes.

The adversarial examples generation algorithm in this paper can be divided into two main steps:

(1) determine the vulnerable tokens to change (2) replace them with the spelling mistakes that most likely to occur in the contexts (Algorithm 1).

For the i-th position of input sentence X, the positional score s_i can be obtained by the logit output o_i as follows:

$$s_i = o_i^{y_i} - o_i^{m_i}(o_i^{m_i} = max\{o_i^r, r \neq y_i\}) \quad (1)$$

where o_i^r denotes the logit output of character r in the i-th position, and y_i denotes the i-th character of ground truth sentence Y. The lower the positional score, the less confident the model is in predicting the position. Attacking this position makes the model output more likely to change. Once the positional score of each character in the input sentence is calculated, we sort these positions in ascending order according to the positional scores. This process can reduce the substitutions and maintain the original semantics as much as possible.

Once a vulnerable position is determined, the token at that position is replaced with one of its phonologically or visually similar characters. Confusion set D contains a set of visually or phonologically similar characters. In order to fool the target CSC model while maintaining the context, the character with the highest logit output in the confusion set is used as a replacement.

Given a sentence in training sets, its adversarial examples are generated by substituting a few characters based on the algorithm mentioned above. Adversarial training was conducted with these examples, improving the robustness of CSC models by alleviating their weak spots, and exploiting knowledge about easy spelling mistakes from confusion sets to help models generalize better.

Algorithm 1 Adversarial Attack Algorithm

Input:
$X = \{x_1, x_2, \ldots, x_n\}$, input Chinese sentence;
$Y = \{y_1, y_2, \ldots, y_n\}$, the corresponding ground truth;
λ, proportion of characters can be changed;
f, a target CSC model;
D, a confusion set created based on visually or phonologically similar characters;
Output:
$\widehat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$, adversarial example;

$\widehat{X} \leftarrow X$
if $f(X) \neq Y$ **then**
  return $\widehat{X}$
else
  $num \leftarrow 0$
  while $f(\widehat{X}) = Y$ && $num \leq \lambda \cdot n$ **do**
    $O = \{o_1, o_2, ..., o_n\} \leftarrow$ Logit output of $f(\widehat{X})$
    $P = \{p_1, p_2, ..., p_k\} \leftarrow$ Sort the position p_i in ascending order based on s_{p_i} ($1 \leq p_i \leq n$ and y_{p_i} is a Chinese character)
    for each $i \in [1, k]$ **do**
      if $\hat{x}_{p_i} \neq y_{p_i}$ **then**
        continue
      end if
      $\hat{x}_{p_i} \leftarrow m_{p_i}$, where $m_{p_i} \in D(x_{p_i})$ and $p_i^{m_{p_i}} = max\{p_i^r, p_i^r \in D(x_{p_i})\}$
      break
    end for
    $num \leftarrow num + 1$
  end while
end if
return $\widehat{X}$

3 Experiments

3.1 Datasets

Statistics of the datasets used are shown in Table 1.

Pretraining data We generated a large corpus by a character substitution-based method. Models were first pre-trained on these nine million sentence pairs, and then fine-tuned using the training data mentioned below.

Training data The training data contained three human-annotated training datasets, SIGHAN 2013 (Wu et al., 2013), SIGHAN 2014 (Yu et al., 2014), and SIGHAN 2015 (Tseng et al., 2015). We also utilized an automatically generated dataset (Wang et al., 2018).

Table 1: Statistics information on the used data resources. A subset of the Wikipedia corpus and Weibo corpus, denoted by Wikipedia* and Weibo* respectively, was sampled from the entire corpus.

Pre-Training Data	#Line	Avg. Length
Wikipedia*	4,531,007	40.2
Weibo*	4,770,015	16.3

Training Data	#Line	Avg. Length	#Errors
(Wang et al., 2018)	271,329	42.6	381,962
SIGHAN 2013	350	49.3	339
SIGHAN 2014	3,437	49.6	5,136
SIGHAN 2015	2,339	31.3	3,048

Test Data	#Line	Avg. Length	#Errors
SIGHAN 2013	1,000	74.3	1,221
SIGHAN 2014	1,062	50.0	771
SIGHAN 2015	1,100	30.7	705

Test data Models' performance in detection and correction stage was evaluated in sentence level on three benchmark datasets, in the metrics of F1 scores (detection and correction). Characters in these datasets were transferred into simplified Chinese characters using OpenCC[2]. We revised the processed datasets for one simplified Chinese character may correspond to multiple traditional Chinese characters.

[2]https://github.com/BYVoid/OpenCC

Table 2: Performance of three models trained with the proposed pretraining strategy and adversarial training method. "**CLEAN**" stands for the testing results on the clean data, and "**ATTACK**" denotes the F1 scores under test-time attacks. "**DET**" and "**COR**" denote the F1 scores of detection and correction. The F1 scores were increased 4.1% on average by our pre-training method across the various models on the different datasets. Models' robustness was also improved about 3.9% while without suffering too much loss (less than 1.1%) on the clean data.

	SIGHAN-2013				SIGHAN-2014				SIGHAN-2015			
	CLEAN		ATTACK		CLEAN		ATTACK		CLEAN		ATTACK	
Model	**DET**	**COR**	**DET**	**COR**	**DET**	**COR**	**DET**	**COR**	**DET**	**COR**	**DET**	**COR**
BERT	82.9	82.1	33.6	15.8	66.8	65.0	41.7	19.0	76.3	74.4	25.1	13.7
+ Pre-trained for CSC	**84.9**	**84.4**	48.5	29.6	**70.4**	**68.6**	51.4	32.4	79.8	78.0	39.0	26.9
+ Adversarial training	84.0	83.5	**50.8**	**31.3**	68.4	66.8	**54.9**	**38.0**	**80.0**	**78.2**	**45.9**	**36.0**
SpellGCN	80.8	80.0	25.6	22.6	64.8	63.6	29.0	24.3	73.6	71.5	18.8	17.4
+ Pre-trained for CSC	**84.6**	**84.0**	28.8	25.8	**67.3**	**66.4**	35.4	27.1	79.6	77.7	26.2	25.2
+ Adversarial training	83.4	82.6	**30.2**	**26.0**	66.4	65.4	**35.9**	**29.5**	**79.6**	**77.8**	**28.2**	**25.2**
Soft-masked BERT	80.6	79.1	27.7	4.0	62.2	59.6	29.8	7.1	72.4	69.6	15.5	5.3
+ Pre-trained for CSC	**84.9**	**84.2**	27.3	6.0	**67.2**	**65.6**	30.7	8.6	**77.2**	**74.5**	22.2	6.5
+ Adversarial training	84.1	83.3	**32.5**	**8.1**	65.0	62.7	**40.5**	**13.4**	76.2	73.8	**30.3**	**11.4**

3.2 Models and Hyper-parameter Settings

For BERT and Soft-Masked BERT, we used the BERT model pre-trained on Chinese text provided by transformers[3] and fine-tuned it. Adam optimizer was used and the learning rate was 2e-5, except when adversarial training on SIGHAN 13 dataset, which was 1e-5. We followed Zhang et al. (2020) to set our hyper-parameters. The size of the hidden state in Bi-GRU in Soft-Masked BERT was 256.

Similarly, we followed the hyper-parameters settings of SpellGCN (Cheng et al., 2020b) except the batch size. Batch size was reduced to eight due to GPU memory. The BERT model used in SpellGCN was provided by the repository of BERT[4].

We conducted adversarial training on base models gained through pre-training and fine-tuning. The threshold λ was tuned on the validation set for each dataset. The number of sentence pairs directly used for training was twice that that used to generate adversarial examples.

3.3 Results and Analysis

As shown in Table 2, through pre-training particularly designed for CSC, the models achieve better results on three benchmark datasets. The average improvement of correction F1 score was 4.3% over base CSC models, which proves that our pre-training method has significant contribution to improving the model. Notably, BERT achieves state-of-the-art results on three datasets through our method.

[3]https://github.com/huggingface/transformers

[4]https://github.com/google-research/bert

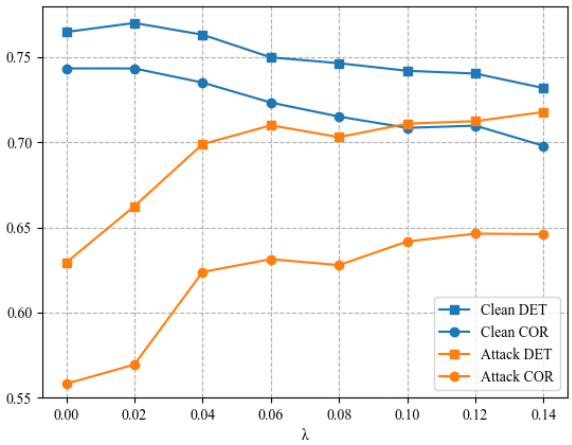


Figure 1: Trade-off between generalization and robustness. The blue and orange lines respectively denote the average F1 scores of BERT on the SIGHAN-2015 data set and the adversarial examples generated ($\lambda = 0.05$).

Figure 1 shows the trade-off between generalization and robustness during adversarial training. As the threshold increases, the robustness of BERT also increases with a slight performance decrease on clean dataset (less than 0.7%).

The experiments of the models under adversarial attacks were conducted with the base, pre-trained and adversarially trained models ($\lambda = 0.02$). We found that CSC models are vulnerable to adversarial examples as expected. The average drop in F1 score of three base models was 51.6%. Under the attacks, the F1 scores of adversarially trained model decreased less (44.1%), which indicates the adversarial training can substantially improve the robustness of CSC models. Compared with other models, BERT is more robust against adversarial attack (-41.2%). The reason for the more serious

robustness issues of other models may be related to the modules added to BERT, which increases the number of parameters, therefore it is more likely to overfit on the CSC data set.

4 Conclusion

In this paper, we have described a character substitution-based method to create large pseudo data to pre-train the models by encouraging them to explore unseen misspellings. We also proposed a data augmentation method for training the CSC models by continually adding the adversarial examples, particularly generated to alleviate the weak spot of the current model, to the training set. By the proposed pre-training strategy and adversarial training method, we can pursue both the exploration and exploitation when training the CSC models. Experimental results demonstrate that the CSC models trained with the data augmented by these pseudo data and adversarial examples can substantially be improved in both generalization and robustness.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by National Key R&D Program of China (No. 2018YFC0830900), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0103), National Science Foundation of China (No. 62076068) and Zhangjiang Lab.

References

Chao-Huang Chang. 1995. A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, volume 95, pages 278–283. Citeseer.

Yong-Zhi Chen, Shih-Hung Wu, Ping-Che Yang, Tsun Ku, and Gwo-Dong Chen. 2011. Improve the detection of improperly used chinese characters in students' essays with error model. *International Journal of Continuing Engineering Education and Life Long Learning*, 21(1):103–116.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020a. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3601–3608.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020b. SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. FASPell: A fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 160–169, Hong Kong, China. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to SIGHAN 2015 bake-off for Chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 32–37, Beijing, China. Association for Computational Linguistics.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for Chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785, Florence, Italy. Association for Computational Linguistics.

Lihao Wang and Xiaoqing Zheng. 2020. Improving grammatical error correction models with purpose-built adversarial examples. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2858–2869, Online. Association for Computational Linguistics.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese spelling check evaluation at SIGHAN bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 35–42, Nagoya, Japan. Asian Federation of Natural Language Processing.

Yang Xin, Hai Zhao, Yuzhu Wang, and Zhongye Jia. 2014. An improved graph model for Chinese spell checking. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 157–166, Wuhan, China. Association for Computational Linguistics.

Jui-Feng Yeh, Sheng-Feng Li, Mei-Rong Wu, Wen-Yi Chen, and Mao-Chuan Su. 2013. Chinese word spelling correction based on n-gram ranked inverted index list. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 43–48, Nagoya, Japan. Asian Federation of Natural Language Processing.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 220–223, Wuhan, China. Association for Computational Linguistics.

Liang-Chih Yu, Lung-Hao Lee, Yuen-Hsien Tseng, and Hsin-Hsi Chen. 2014. Overview of SIGHAN 2014 bake-off for Chinese spelling check. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 126–132, Wuhan, China. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 882–890, Online. Association for Computational Linguistics.

Shuiyuan Zhang, Jinhua Xiong, Jianpeng Hou, Qiao Zhang, and Xueqi Cheng. 2015. HANSpeller++: A unified framework for Chinese spelling correction. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 38–45, Beijing, China. Association for Computational Linguistics.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.

Training Adaptive Computation for Open-Domain Question Answering with Computational Constraints

Yuxiang Wu **Pasquale Minervini** **Pontus Stenetorp** **Sebastian Riedel**
University College London
{yuxiang.wu,p.minervini,p.stenetorp,s.riedel}@cs.ucl.ac.uk

Abstract

Adaptive Computation (AC) has been shown to be effective in improving the efficiency of Open-Domain Question Answering (ODQA) systems. However, current AC approaches require tuning of all model parameters, and training state-of-the-art ODQA models requires significant computational resources that may not be available for most researchers. We propose *Adaptive Passage Encoder*, an AC method that can be applied to an existing ODQA model and can be trained efficiently on a single GPU. It keeps the parameters of the base ODQA model fixed, but it overrides the default layer-by-layer computation of the encoder with an AC policy that is trained to optimise the computational efficiency of the model. Our experimental results show that our method improves upon a state-of-the-art model on two datasets, and is also more accurate than previous AC methods due to the stronger base ODQA model. All source code and datasets are available at https://github.com/uclnlp/APE.

1 Introduction

Open-Domain Question Answering (ODQA) requires finding relevant information for a given question and aggregating the information to produce an answer. The retriever-reader architecture, popularised by Chen et al. (2017), has shown great success in this task. The retriever acquires a set of documents from external sources (e.g., Wikipedia) and the reader extracts the answer spans from these documents (Clark and Gardner, 2018; Yang et al., 2019; Wang et al., 2019; Min et al., 2019; Asai et al., 2020). Recently, Min et al. (2020); Lewis et al. (2020b); Izacard and Grave (2020b) showed that generative reader models that exploit an encoder-decoder architecture can significantly outperform previous extractive models, thanks to their better capability in aggregating and combining evidence from multiple passages. However, these generative models are much more computationally expensive than extractive models, and often need to be trained with a large number of passages, making it hard to train these models for most researchers (Schwartz et al., 2020a).

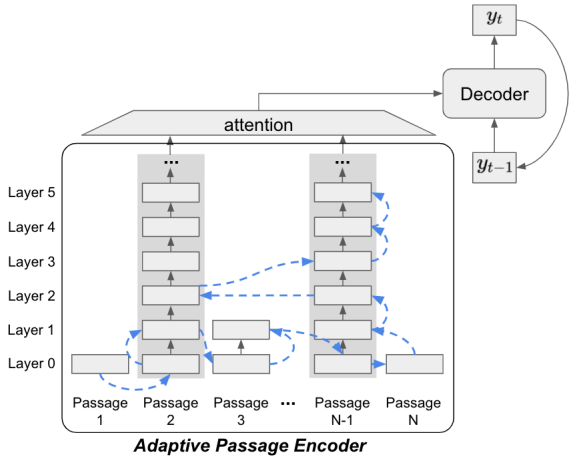


Figure 1: Overview of our approach. The adaptive passage encoder overrides the layer-by-layer computation of the encoder with an adaptive computation policy (indicated in blue dash arrows).

Wu et al. (2020) show that Adaptive Computation (AC) can significantly improve the efficiency of extractive ODQA models at inference time. However, it requires fine-tuning all model parameters with a multitask learning objective, making it computationally challenging to apply this method to current state-of-the-art models.

In this work, we explore an efficient approach to apply adaptive computation to large generative ODQA models. We introduce the *Adaptive Passage Encoder* (APE), a module that can be added to the encoder of an existing ODQA model, which has the following features: *1*) it efficiently reuses the encoder's hidden representations for calculating

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 447–453
August 1–6, 2021. ©2021 Association for Computational Linguistics

the AC priorities; *2*) it does not require tuning of the base model and hence allows efficient training under limited resource; *3*) it does not require confidence calibration. Our experimental results on NaturalQuestions and TriviaQA show that our method improves the performance of the state-of-the-art model FiD (Izacard and Grave, 2020b), while also producing more accurate results (12.4% EM) than the AC method proposed by Wu et al. (2020).

2 Related Work

Open Domain Question Answering ODQA is a task that aims to answer a factoid question given a document corpus. Most works in this domain follow a *retriever-reader* design first proposed by Chen et al. (2017). The retriever collects a set of relevant passages, then the reader comprehends and aggregates the information from multiple passages to produce the answer. Depending on the design of the reader model, these systems could be further categorised into *extractive models* and *generative models*. Extractive models (Min et al., 2019; Yang et al., 2019; Wang et al., 2019; Asai et al., 2020; Karpukhin et al., 2020) exploit an answer extraction model to predict the probabilities of answer spans, and use global normalisation (Clark and Gardner, 2018) to aggregate the answer probabilities across multiple passages.

However, thanks to recent advances in sequence-to-sequence pretrained language models (Raffel et al., 2020; Lewis et al., 2020a), generative ODQA models (Min et al., 2020; Lewis et al., 2020b; Izacard and Grave, 2020b) achieve significant improvement upon extractive models, demonstrating stronger capability in combining evidence from multiple passages. We focus on generative models in this work.

Passage Retrieval and Re-Ranking Passage retrievers in ODQA systems are initially based on sparse vector representations. Chen et al. (2017) use TF-IDF, whereas Yang et al. (2019); Karpukhin et al. (2020); Wang et al. (2019) rely on BM25 for ranking passages (Robertson, 2004). Recently, Karpukhin et al. (2020); Lewis et al. (2020b); Izacard and Grave (2020a) achieved substantial increase in retrieval performance using dense representations. Our work is based on the retrieval results from a dense retriever (Izacard and Grave, 2020b), but we show that the proposed method can still improve the quality of the support passages despite the strong retrieval performance.

Nogueira and Cho (2019); Qiao et al. (2019); Mao et al. (2021) show that adding a separate cross-encoder re-ranker can improve the performance, but that comes with a significant increase of the computation at train or inference time. Despite that our proposed adaptive passage encoder can be viewed as an encoder with an integrated re-ranker, the focus of our work is to improve the computational efficiency, namely, enhancing the performance without a substantial increase in computation.

Adaptive Computation Adaptive computation allows the model to condition the computation cost on the input. For example, Schwartz et al. (2020b); Liu et al. (2020); Xin et al. (2020) propose models that can dynamically decide to early exit at intermediate layers when the confidence at the layer exceeds a threshold. They show that adaptively early exiting can significantly reduce the computational cost for various sequence classification tasks. Closest to our work, Wu et al. (2020) introduced adaptive computation for extractive ODQA models. We extend adaptive computation to generative ODQA models, and our approach can be incorporated in existing generative ODQA models without finetuning the base model.

3 Method

In this section, we will introduce the base model and how our proposed adaptive passage encoder works with it.

3.1 Base Model

Large generative ODQA models (Lewis et al., 2020b; Izacard and Grave, 2020b) share a similar encoder-decoder architecture. They first concatenate the question with all retrieved passages. Then the encoder encodes all passages and produces their hidden representations $h_1^L, \cdots, h_N^L$, where L is the number of encoder layers and N is the number of retrieved passages. We denote the hidden representation of the i-th passage at its j-th encoder layer as h_i^j. The decoder will attend to these hidden representations and generate the answer tokens sequentially.

3.2 Adaptive Passage Encoder

As shown in Fig. 1, the adaptive passage encoder overrides the layer-by-layer computation of the encoder of the base model with an adaptive computation policy. It adds two components on top of the

base encoder to define the policy: an answerability prediction model HasAnswer and a scheduler.

The HasAnswer model predicts the probability that a passage contains an answer to the question, given its hidden representation h_i^j. It first pools hidden representation h_i^j into a vector, then feeds the pooled representation to a multi-layer perceptron to produce the probability p_i^j.

The scheduler is then responsible for the selection and prioritisation of passages that are likely to contain the answer (Wu et al., 2020). As shown by the blue arrows in Fig. 1, the scheduler learns a scheduling policy to allocate encoder layer computation to passages. The scheduler will exit in early layers for those spurious passages while allocating more layers to the ones that it finds promising.

To achieve this goal, the scheduler produces a *priority* score q_n for each passage:

$$q_n = \sigma(g(p_n^{l_n}, n, l_n))p_n^{l_n} + f(p_n^{l_n}, n, l_n) \quad (1)$$

where n is the passage rank by the retriever, l_n is the index of its current encoder layer, g and f are two multi-layer perceptrons that learn the weight and bias respectively. Starting at the initial layer for all passages, the scheduler will select a passage with the maximum priority, forward one encoder layer for it $l'_n = l_n + 1$, and updates its priorities q_n with its new hidden representation $h_n^{l'_n}$ and has-answer probability $p_n^{l'_n}$. This process will iterate for B (budget) steps, and only k passages with the most layers computed are retained in the end.

3.3 Training the Adaptive Passage Encoder

Differently from Wu et al. (2020), our method does not require tuning the underlying base model. Since the number of parameters introduced by the HasAnswer model and the scheduler is less than 4% of the base model, APE can be trained very efficiently. The HasAnswer model is first trained with cross-entropy loss, supervised by the has-answer labels of the passages. Then we fix HasAnswer and train the scheduler with REINFORCE algorithm (Williams, 1992) to maximise the expected return, which is defined to encourage selection and prioritisation of passages that contain the answer. The selection action gains a positive reward $(1 - c)$ if it selects a relevant passage, otherwise a negative reward $-c$. Since the weight g and bias f in Eq. (1) are automatically learned during the training of the scheduler, our method does not require confidence calibration of the HasAnswer model, unlike the method proposed by Wu et al. (2020).

	Train	Validation	Test
NaturalQuestions	79,168	8,757	3,610
TriviaQA	78,785	8,837	11,313

Table 1: Number of samples of the evaluated datasets.

4 Experiments

4.1 Experimental Setup

Datasets Following (Lee et al., 2019; Izacard and Grave, 2020b), we evaluate our method on NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) whose statistics are shown in Table 1.

Evaluation Metrics Following Wu et al. (2020), we conduct the evaluation under different computational costs at inference time. Since the number of passages k is almost linearly correlated with memory consumption and number of operations, we evaluate the performances with various number of passages $k \in \{5, 10, 20\}$. To evaluate the end performance of ODQA models, we use the standard Exact Match (EM) score, which is the proportion of questions whose predicted answer matches exactly with the ground truth. We also include the unrestricted setting to compare the best performances of different models.

Technical Details We use FiD (Izacard and Grave, 2020b) as our base model. FiD-base and FiD-large contain $L = 12$ and 24 layers respectively, and we set the budget $B = Lk$. For the pooling operation in the HasAnswer model, we found max-pooling works better than mean-pooling and the [CLS] token, so max-pooling is used in all our experiments. We use discount factor $\gamma = 0.8$ and step penalty $c = 0.1$ during the REINFORCE training of the scheduler. More hyperparameters are presented in Appendix A.1.

Computational Feasibility Tuning a FiD-base model with $k = 20$ or a FiD-large model with $k = 10$ (batch size=1) would yield out-of-memory errors on a V100 (16GB) GPU. Hence, it is infeasible to train FiD with the previous AC method (Wu et al., 2020) in our setting. However, training with our proposed approach can be done in the same setting with a batch size 4 or larger within 8-15

	NaturalQuestions				TriviaQA			
	Top-5	Top-10	Top-20	Unrestricted	Top-5	Top-10	Top-20	Unrestricted
SkylineBuilder (Wu et al., 2020)	34.4	34.2	-	34.2	-	-	-	-
DPR (Karpukhin et al., 2020)	-	40.8	-	41.5	-	-	-	57.9
DPR (our implementation)	38.4	40.2	40.2	40.2	-	-	-	-
RAG (Lewis et al., 2020b)	**43.5**	44.1	44.1	44.5	-	-	-	56.1
FiD-base (Izacard and Grave, 2020b)	39.5	42.9	45.3	48.2	53.9	57.9	60.7	65.0
Ours (APE+FiD-base)	**40.3**	**43.7**	**46.0**	48.2	**55.4***	**59.0***	**62.0***	65.0
FiD-large (Izacard and Grave, 2020b)	42.5	45.8	48.3	51.4	57.2	60.6	63.7	67.6
Ours (APE+FiD-large)	**43.4**	**46.6**	**49.1**	51.4	**57.9**	**61.4***	**64.1***	67.6

Table 2: Exact match scores on NaturalQuestions and TriviaQA test sets. * indicates statistical significance.

	NaturalQuestions				TriviaQA			
	Top-5	Top-10	Top-20	Top-100	Top-5	Top-10	Top-20	Top-100
BM25 (Lee et al., 2019)	-	-	59.1	73.7	-	-	66.9	76.7
DPR (Karpukhin et al., 2020)	67.1	-	78.4	85.4	-	-	79.4	85.0
FiD (Izacard and Grave, 2020b)	66.2	73.9	79.2	86.1	69.8	74.9	78.9	84.8
Ours (APE+FiD-base)	**67.4***	75.1*	**80.4***	86.1	**70.8***	**75.8***	**79.5**	84.8
Ours (APE+FiD-large)	67.2	**75.4***	80.2*	86.1	70.4	75.6*	79.2	84.8

Table 3: Top-k retrieval accuracy scores on NaturalQuestions and TriviaQA test sets. * indicates statistical significance.

hours.

4.2 Experimental Results

As shown in Table 2 under restricted top-k, our proposed method improves upon the FiD model on both datasets, and by a statistically significant margin on TriviaQA. It also outperforms the previous AC method (Wu et al., 2020) by 12.4% when $k = 10$ due to the stronger base model. The addition of APE allows FiD to significantly outperform RAG (Lewis et al., 2020b) on NaturalQuestions when $k \in \{10, 20\}$.

Previous adaptive computation methods (Wu et al., 2020; Schwartz et al., 2020b) was reported to have plateaued or degraded performances in the unrestricted setting. However, Table 2 shows that our approach does not have this issue.

4.3 Analysis of Passage Quality

To understand how APE outperforms the baselines, we analyse the quality of the final top-k passages retained by APE. Table 3 reports the top-k retrieval accuracy of the top-k passages. The results show that the top-k accuracy of the selected collection of documents by APE is significantly better than BM25, DPR, and FiD, which are strong retrieval baselines for ODQA. Combined with Table 2, it indicates that the better passage quality yielded by APE helps to improve the end ODQA performance of the model.

5 Conclusions

In this work, we explore an adaptive computation method that can be efficiently applied to an existing generative ODQA model. We find that, by replacing the encoder of generative ODQA models with our proposed adaptive passage encoder, we can train an effective adaptive computation policy without tuning the base model. This allows applying adaptive computation to large state-of-the-art generative models, which was previously challenging computation-wise. Our experimental results show that our method produces more accurate results than a state-of-the-art generative model on both NaturalQuestions and TriviaQA, and it outperforms the previous AC method by a large margin. The analysis also shows that our approach achieves better passage quality that leads to improvements in ODQA performance.

Acknowledgments

The first author would like to thank his wife Jane for her love and support throughout the years. We would also like to thank Gautier Izacard and Edouard Grave for their help with using FiD. This research was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 875160.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *ICLR*. OpenReview.net.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL (1)*, pages 1870–1879. Association for Computational Linguistics.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *ACL (1)*, pages 845–855. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2020a. Distilling knowledge from reader to retriever for question answering. *CoRR*, abs/2012.04584.

Gautier Izacard and Edouard Grave. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *CoRR*, abs/2007.01282.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL (1)*, pages 1601–1611. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *ACL (1)*, pages 6086–6096. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. Fastbert: a self-distilling BERT with adaptive inference time. In *ACL*, pages 6035–6044. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Reader-guided passage reranking for open-domain question answering. *CoRR*, abs/2101.00294.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *EMNLP/IJCNLP (1)*, pages 2851–2864. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *EMNLP (1)*, pages 5783–5797. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of BERT in ranking. *CoRR*, abs/1904.07531.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020a. Green AI. *Commun. ACM*, 63(12):54–63.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020b. The right tool for the job: Matching model and instance complexities. In *ACL*, pages 6640–6651. Association for Computational Linguistics.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *EMNLP/IJCNLP (1)*, pages 5877–5881. Association for Computational Linguistics.

R. J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Yuxiang Wu, Sebastian Riedel, Pasquale Minervini, and Pontus Stenetorp. 2020. Don't read too much into it: Adaptive computation for open-domain question answering. In *EMNLP (1)*, pages 3029–3039. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating BERT inference. In *ACL*, pages 2246–2251. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *NAACL-HLT (Demonstrations)*, pages 72–77. Association for Computational Linguistics.

A Experimental Details

A.1 Hyper-parameters

Hyper-parameter	Value
learning rate	1e-4
batch size	24
epoch	2
optimiser	Adam
Adam ϵ	1e-6
Adam (β_1, β_2)	(0.9, 0.999)
max sequence length	256
pooling	max-pooling
number of passages	5/10/20
device	Nvidia V100

Table 4: Hyper-parameters for the HasAnswer model training.

Hyper-parameter	Value
learning rate	0.01
batch size	24
epoch	1
optimiser	Adam
max number of steps	240
step cost c	0.1
discount factor γ	0.8
hidden size of MLPs	64
number of passages	20/30/50

Table 5: Hyper-parameters for scheduler model REINFORCE training.

An Empirical Study on Adversarial Attack on NMT: Languages and Positions Matter

Zhiyuan Zeng[1] and Deyi Xiong[2] *
School of New Media and Communication, Tianjin University, Tianjin, China[1]
College of Intelligence and Computing, Tianjin University, Tianjin, China[2]
{zhiyuan_zeng, dyxiong}@tju.edu.cn

Abstract

In this paper, we empirically investigate adversarial attack on NMT from two aspects: languages (the source vs. the target language) and positions (front vs. back). For autoregressive NMT models that generate target words from left to right, we observe that adversarial attack on the source language is more effective than on the target language, and that attacking front positions of target sentences or positions of source sentences aligned to the front positions of corresponding target sentences is more effective than attacking other positions. We further exploit the attention distribution of the victim model to attack source sentences at positions that have a strong association with front target words. Experiment results demonstrate that our attention-based adversarial attack is more effective than adversarial attacks by sampling positions randomly or according to gradients.

1 Introduction

Despite remarkable progress in recent years, neural machine translation (NMT) models are vulnerable to small perturbations (Cheng et al., 2018; Zhao et al., 2018). Adversarial training, which allows NMT models to learn from adversarial samples with perturbations, as a general approach, is widely used to improve the robustness of NMT (Ebrahimi et al., 2018; Vaibhav et al., 2019; Cheng et al., 2019, 2020a,a; Zou et al., 2019). Generally, NMT models yield target translations in an autoregressive way[1], which makes previous incorrectly predicted target tokens have a negative impact on future tokens to be generated. However, most approaches to generating NMT adversarial examples inject perturbations only into source sentences. Hence, are NMT models more vulnerable to adversarial attack on the source side? What roles do injecting perturbations into source sentences or into target translations play in improving the robustness of NMT?

The key interest of this paper is to attempt to answer these questions by an empirical and comparative study on different adversarial attacks on NMT models. First, we investigate adversarial attacks on the source side versus those on the target side. This study is to know which attack is more effective for NMT by measuring performance drop of the attacked models. Second, we empirically study the impact of attacking different positions on either source sentences or target translations to find whether NMT robustness is sensitive to positions. Third, based on the findings of the study, we propose a new adversarial attack generation method based on attention distribution.

Our contributions can be summarized as follows:

- By the study, we have empirically found that adversarial attack on the source side is more effective than that on the target side in terms of the performance degradation of NMT models under attack.
- We have further empirically found that adversarial attacks on front positions are more effective than those on back positions on the target side due to the autoregressive translation nature. We have also found that adversarial attacks on positions of the source side which are aligned to front positions of the target side are more effective than attacks on other positions on the source side.
- We propose a new adversarial attack generation approach that samples positions for injecting perturbations according to the attention distribution. Experiment results demonstrate that attention-based position sampling is more effective than random sampling and gradient-

*Corresponding author

[1]We leave the study of adversarial attack to non-autoregressive NMT models to our future work.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 454–460
August 1–6, 2021. ©2021 Association for Computational Linguistics

based sampling.

2 Related Work

Robustness is a well-known problem for neural networks (Szegedy et al., 2014; Goodfellow et al., 2015). Recent years have witnessed that many adversarial training approaches have been proposed to improve the robustness of NMT models. Cheng et al. (2018) generate adversarial samples at the lexical and feature level, and apply the adversarial learning to make adversarial samples natural. Zhao et al. (2018) utilize generative adversarial networks to generate adversarial examples that lie on the data manifold by searching in the semantic space of dense and continuous data representations. Ebrahimi et al. (2018) propose an attack framework for character-level NMT, which uses gradient to rank adversarial manipulations and to search for adversarial examples via either greedy search or beam search methods. Belinkov and Bisk (2018) attack character-level NMT by randomizing the order of letters or randomly replacing letters with their adjacent letters on the keyboard. Vaibhav et al. (2019) use back translation to generate adversarial samples that emulate natural noises. Cheng et al. (2020a) exploit the projected gradient method combined with gradient regularization to generate adversarial samples. Zou et al. (2019) employ reinforcement learning to decide which positions to attack. Tan et al. (2020) present a method to change inflectional morphology of words to craft plausible and semantically similar adversarial examples. Emelin et al. (2020) propose to generate adversarial examples by eliciting disambiguation errors.

All these approaches attack the source side of NMT in different ways. However distortions exist in not only the source language, but also the target language. This inspires us to compare the effectiveness of adversarial attack on the source and target side to NMT models. We have found that the NMT models are vulnerable to both the source and target attack. However, to our best knowledge, only Cheng et al. (2019) and Cheng et al. (2020b) take noises in target sentences into account. They generate adversarial samples for both source and target sentences. Their target-side adversarial examples are generated according to the attacked positions in corresponding source sentences, while their source-side adversarial samples are generated by randomly sampling positions to attack. We improve their method by attacking the source side according to the attention distribution. Experiments validate the effectiveness of our method.

3 Data and Setup

We conducted experiments on two translation tasks: English-Chinese and English-Japanese. Data for English-Chinese translation are from the United Nations English-Chinese corpus (Ziemski et al., 2016). We built the training/validate/test set for this task by randomly sampling 3M/2K/2K sentence pairs from the whole corpus. For the English-Japanese translation task, we aggregated the training set of KFTT (Neubig, 2011), JESC (Pryzant et al., 2018) and TED talks (Cettolo et al., 2012) as our training set, which consists of 3.9M sentence pairs. We evaluated our models on the validation set and test set of KFTT (Neubig, 2011). We split words into sub-word units with subword regularization (Kudo, 2018) and built a shared vocabulary of 32K subwords for both English-Chinese and English-Japanese.

We used the base Transformer model (Vaswani et al., 2017) with 512 hidden units as the victim model. The hyper-parameters of the base Transformer follows the default setting in Vaswani et al. (2017). We implemented the adversarial attack and training methods of Cheng et al. (2019) and followed their hyper-parameter setting in our experiments. The details of our implementation is shown in Section 4.

We injected perturbations into either source sentences or target sentences to generate adversarial examples which were used to evaluate NMT models. Since we could not inject perturbations into the target inputs of NMT models at the test time, we evaluated NMT models with target-side adversarial samples at training time on the validation dataset. Except where otherwise specified, the performance of the victim model was measured by word accuracy on the validation data. If we evaluated the victim model on the test set, detokenized BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020) were reported. Although the target-side inputs of NMT models could not be attacked at test time, there still exists noise or errors in them due to error propagation in the autoregressive decoding. Evaluating NMT with perturbated target sentences at training time enables us to analyze the vulnerability of NMT to the noise in target-side inputs, and inspires us to improve the robustness of NMT models to such noise.

src-tgt	en-zh	zh-en	en-ja	ja-en
noisy-clean	**55.79**	**61.89**	**50.21**	**51.98**
clean-noisy	61.32	64.00	52.74	52.58
clean-clean	71.16	78.76	60.35	62.30
noisy-noisy	46.55	46.55	40.63	40.78

Table 1: Word Translation accuracy of victim model under the adversarial attack on the source (src) vs. target (tgt)

4 Implementation Details

The adversarial attack and training framework used in this paper is based on Cheng et al. (2019). They inject perturbations into the source/target sentences by replacing a word in a sentence with the words that are semantically similar to the words being replaced. Words to be replaced in a source sentence are sampled according to the uniform distribution, while those in a target sentence are sampled according to the attention distribution. We tried three different ways to sample words to inject perturbations into source sentence in Section 8. Given a word to be replaced, Cheng et al. (2019) use a bi-directional language model to choose candidate words from vocabulary which share similar semantics to it, and then use gradients to search a word from candidate words to replace it. Cheng et al. (2019) combine a left-to-right and right-to-left language model to rank candidate words, while we combine the two uni-directional language models by multiplying their likelihood for simplicity. Cheng et al. (2019) train their NMT models with both clean data and adversarial samples from scratch. To save training time, we pretrain our NMT models with clean data before adversarial training.

5 Adversarial Attack on Source vs. Target

In this section, we compare the effect of the source and target attack according to the performance of the victim model. We adversarially inject perturbations into source sentences and keep target translations unchanged (clean) for the source attack while the target attack works the other way around. Our adversarial examples for both the source and target attack are generated by the method of Cheng et al. (2019). To make the comparison between the source and target attack fair, we randomly sample positions to attack for both of them.

Results are shown in Table 1. The NMT model with noisy source and clean target performs worse

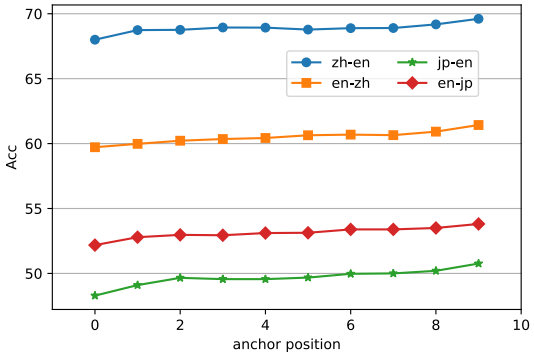


Figure 1: The word translation accuracy of an NMT model under attack at different anchor positions on the target side.

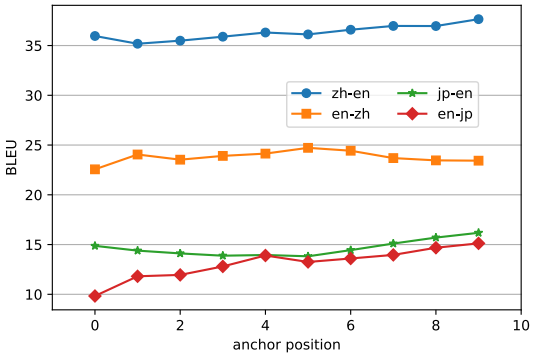


Figure 2: The BLEU score of an NMT model under attack at different anchor positions on the source side.

than that with clean source and noisy target on all translation tasks, in terms of word translation accuracy, which indicates that the source attack is more effective than the target attack. We also oberve that the adversarial attack on the source together with target side is much better than that on a single side, therefore we suggest that adversarial attacks on both the source and target side should be conducted to deploy a robust NMT system.

6 Adversarial Attack at Different Positions

In this section, we investigate the impact of attacked positions in the source and target sentences on NMT. We start with adversarial attack on the target side. Adversarial attacks at the front of a target sentence are supposed to be more effective than those at the end of the target sentence, since noises in the front of the target sentence will negatively affect future target tokens, while noises at the end of the target sentence could not affect already

generated tokens for a left-to-right decoder.

Given a sentence of length L, we uniformly select 10 anchor positions from the sentence:

$$\hat{x_j} = [\frac{L \times j}{10}] \quad (1)$$

where $\hat{x_j}$ is the jth anchor position ($0 <= j < 10$), $[\cdot]$ is a rounding operation. For each anchor position $\hat{x_j}$, we sample several positions close to it according to the discrete Gaussian distribution, which is formulated as:

$$p(x) = \frac{e^{-(x-\hat{x})^2}}{\sum_{i=0}^{L-1} e^{-(i-\hat{x})^2}} \quad (2)$$

where $p(x)$ is the probability that position x is attacked, $\hat{x}$ is the anchor position that we want the sampled positions to surround. The denominator normalizes the sum of the probabilities to 1.

Results of adversarial attack on different anchor positions on the target side are shown in Figure 1. On all translation tasks, the word translation accuracy of the victim model goes up as attacked positions move from the starting position to the end of target sentences, which confirms that attacking at the front of a target sentence is more effective than attacking at the end.

We also perform adversarial attack on source sentences at different anchor positions. Results are displayed in Figure 2. We measure the performance of the victim model for the source attack at different positions on the test set with the metric of BLEU. On both English-Chinese and English-Japanese tasks, BLEU scores go up as the attacked positions move from the start to the end of source sentences, which indicates that attacking the front of a source sentence is also more effective than attacking the end for both English-Chinese and English-Japanese translation. We suppose that the reason for this is that words at front positions of source sentences usually align to words at front positions of target sentences for the two language pairs. Experiment results in Section 7 empirically validate this hypothesis.

7 Attention Weights at Different Positions

In section 6, we suppose that words at front positions of source sentences usually align to words at front positions of target sentences for both English-Chinese and English-Japanese. We empirically validate this by comparing the attention weights from source tokens at different positions to the first target token. Following the sampling technique in section 6, we uniformly select 10 anchor positions from a source sentence and then sample positions surrounding these anchor positions according to the distribution formulated in Eq (2). For every anchor position, we report the sum of attention weights from the sampled source tokens around the anchor position to the first target token. The results are shown in Figure 3. As expected, the attention weights from the sampled source tokens to the first target token go down as the their corresponding positions move from the start to the end of the source sentence in both English-Chinese and English-Japanese translation, which confirms that words at front positions of source sentences have a stronger association with words at front positions of target sentences than other positions for the two language pairs.

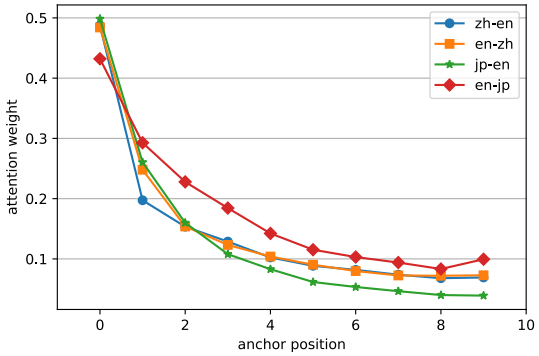


Figure 3: Attention weights from different anchor positions of source sentences to the first token of target sentences.

8 Adversarial Attack based on Attention Distribution

In Section 6, we have found that generating perturbations at front positions on the target side is more effective than attacking other positions. As attention weights in NMT models can be seen as the strength of association between the source and target tokens (Bahdanau et al., 2015). Hence we sample positions of a source sentence to inject perturbations according to the attention distribution. Particularly, the query used to produce the attention distribution is the representation of the first target token and the key is the set of representations of source tokens. There are multiple cross-attention heads in Transformer, each of which produces an

attack / model	rand		grad		attn	
	BELU	BERTScore	BELU	BERTScore	BELU	BERTScore
victim	19.2	83.8	22.2	84.7	**17.4**	**83.4**
train-rand	25.0	86.1	28.9	87.1	**22.2**	**85.6**
train-grad	23.6	85.7	28.6	87.1	**21.3**	**85.2**
train-attn	24.0	85.7	27.7	86.8	**23.3**	85.8

Table 2: BLEU and BERTScore of the victim model and three adversarially trained models. "rand", "grad" and "attn" indicates that adversarial examples are generated at attacked positions sampled randomly, according to gradients and attention distribution, respectively. "train-X" denotes that NMT models are adversarially trained with adversarial examples generated by the "X" method. The models were evaluated on the test set of the English-Chinese corpus.

attention matrix. The average of attention distributions of all heads is hence used for attacking.

We compare our proposed attention-based attack with attacks that either randomly sample source positions or sample positions according to gradients. For gradient-based sampling, we follow Liang et al. (2018) to estimate the L_∞ norm of the gradient of a word embedding as the importance score of the corresponding word, and then sample positions to attack from the normalized importance score. We have implemented the three adversarial attack methods based on the framework proposed in Cheng et al. (2019).[2] The only difference of these methods is that they use different ways to sample positions to attack. We also use the adversarial training method proposed in Cheng et al. (2019) to fine-tune NMT model with adversarial samples generated with the three attacking methods.

BLEU scores and BERTScores of the three adversarially trained models on the test set are shown in Table 2. It can be seen that BLEU scores and BERTScores of almost all models under our proposed attack ("attn") are lower than those under the other two attacking methods, which indicates the superiority of the proposed attention-based attack over the other two attack methods. It is surprising that the attack that samples positions according to the gradient ("grad") is not better than the attack that samples from a uniform distribution ("rand"), which may suggest that the L_∞ norm of the gradient cannot measure the importance of a word in a sentence. We can further extend our method to the black-box attack with the alignment from SMT models (Och and Ney, 2003), which is left to our future work. Our attention-based attack is proposed for autoregressive NMT models that generate target translations from left to right. It will not work for non-autoregressive NMT models (Gu et al., 2017) or autoregressive NMT models that generates translations in an arbitrary order (Stern et al., 2019).

[2]Cheng et al. (2019) randomly sample positions to attack source sentences in their paper.

9 Conclusion

In this paper, we have empirically investigated adversarial attack on NMT models. We compare adversarial attack on the source vs. target side, and find that the former is more effective than the latter. We also study adversarial attack at different positions in either source or target sentences, and observe that attacking front positions in either source or target sentences for English-Chinese and English-Japanese translation is more effective than attacking back positions. We further exploit attention distribution to attack words of a source sentence at positions that have a high association with words at front positions of the corresponding target sentence. Experiments validate the effectiveness of our proposed attention-based attack.

Acknowledgements

The present research was partially supported by OPPO. We would like to thank the anonymous reviewers for their insightful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC,*

Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation, EAMT 2012, Trento, Italy, May 28-30, 2012*, pages 261–268. European Association for Machine Translation.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020a. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3601–3608. AAAI Press.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4324–4333. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020b. Advaug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5961–5970. Association for Computational Linguistics.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1756–1766. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 653–663. Association for Computational Linguistics.

Denis Emelin, Ivan Titov, and Rico Sennrich. 2020. Detecting word sense disambiguation biases in machine translation for model-agnostic adversarial attacks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7635–7653. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *CoRR*, abs/1711.02281.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4208–4215. ijcai.org.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: japanese-english subtitle corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Samson Tan, Shafiq R. Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 2920–2935. Association for Computational Linguistics.

Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1916–1920. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2019. A reinforced generation of adversarial samples for neural machine translation. *CoRR*, abs/1911.03677.

OntoGUM: Evaluating Contextualized SOTA Coreference Resolution on 12 More Genres

Yilun Zhu[1], **Sameer Pradhan**[2,3], and **Amir Zeldes**[1]

[1]Department of Linguistics, Georgetown University
[2]Linguistic Data Consortium, University of Pennsylvania
[3]cemantix.org

yz565@georgetown.edu, pradhan@cemantix.org, Amir.Zeldes@georgetown.edu

Abstract

SOTA coreference resolution produces increasingly impressive scores on the OntoNotes benchmark. However lack of comparable data following the same scheme for more genres makes it difficult to evaluate generalizability to open domain data. This paper provides a dataset and comprehensive evaluation showing that the latest neural LM based end-to-end systems degrade very substantially out of domain. We make an OntoNotes-like coreference dataset called OntoGUM publicly available, converted from GUM, an English corpus covering 12 genres, using deterministic rules, which we evaluate. Thanks to the rich syntactic and discourse annotations in GUM, we are able to create the largest human-annotated coreference corpus following the OntoNotes guidelines, and the first to be evaluated for consistency with the OntoNotes scheme. Out-of-domain evaluation across 12 genres shows nearly 15-20% degradation for both deterministic and deep learning systems, indicating a lack of generalizability or covert overfitting in existing coreference resolution models.

1 Introduction

Coreference resolution is the task of grouping referring expressions that point to the same entity, such as noun phrases and the pronouns that refer to them. The task entails detecting correct mention or 'markable' boundaries and creating a link with previous mentions, or antecedents. A coreference chain is a series of decisions which groups the markables into clusters. As a key component in Natural Language Understanding (NLU), the task can benefit a series of downstream applications such as Entity Linking, Dialogue Systems, Machine Translation, Summarization, and more (Poesio et al., 2016).

In recent years, deep learning models have achieved high scores in coreference resolution. The end-to-end approach (Lee et al., 2017, 2018) jointly scoring mention detection and resolution currently not only beats earlier rule-based and statistical methods but also outperforms other deep learning approaches (Wiseman et al., 2016; Clark and Manning, 2016a,b). Additionally, language models trained on billions of words significantly improve performance by providing rich word and context-level information for classifiers (Lee et al., 2018; Joshi et al., 2019a,b).

However, scores on the identity coreference layer of benchmark OntoNotes dataset (Pradhan et al., 2013) do not reflect the generalizability of these systems. Moosavi and Strube (2017) pointed out that lexicalized coreference resolution models, including neural models using word embeddings, face a covert overfitting problem because of a large overlap between the vocabulary of coreferring mentions in the OntoNotes training and evaluation sets. This suggests that higher scores on OntoNotes-test may not indicate a better solution to the coreference resolution task.

To investigate the generalization problem of neural models, several projects have tested other datasets consistent with the OntoNotes scheme. Moosavi and Strube (2018) conducted out-of-domain evaluation on WikiCoref (Ghaddar and Langlais, 2016), a small dataset employing the same coreference definitions. Results showed that neural models (with fixed embeddings) do not achieve comparable performance (16.8% degradation in score) as on OntoNotes. More recently, the e2e model using BERT (Joshi et al., 2019b) showed gains on the GAP corpus (Webster et al., 2018) using contextualized embeddings; however GAP only contains name-pronoun coreference, a very specific subset of coreference, and is limited in domain to the same single source – Wikipedia.

Though previous work has already identified the overfitting problem, it also has three main shortcomings. First, the scale of out-of-domain evalua-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 461–467
August 1–6, 2021. ©2021 Association for Computational Linguistics

Genre	Documents	Tokens	Mentions	Proper	Pron.	Other	Clusters
academic (ac)	16	15,112	1,232	283	262	687	421
bio (bi)	20	17,963	2,312	934	796	582	487
conversation (cn)	5	5,701	1,027	40	728	259	176
fiction (fc)	18	16,312	2,740	259	1,700	781	469
interview (it)	19	18,060	2,622	501	1,223	898	608
news (nw)	21	14,094	1,803	796	340	667	477
reddit (rd)	18	16,286	2,297	117	1,336	844	578
speech (sp)	5	4,834	601	171	245	185	134
textbook (tx)	5	5,379	466	108	165	193	133
vlog (vl)	5	5,189	882	22	600	260	149
voyage (vy)	17	14,967	1,339	564	300	475	348
whow (wh)	19	16,927	2,057	53	1,001	1,003	491
Total	168	150,824	19,378	3,848	8,696	68,34	4,471

Table 1: Genre-breakdown Statistics of OntoGUM.

tion has been small and homogeneous: WikiCoref only contains 30 documents with ∼60K tokens, much smaller than the OntoNotes test set, and the single genre Wiki domain in both WikiCoref and GAP is arguably not very far from some OntoNotes materials. Second, pretrained LMs, e.g. BERT (Devlin et al., 2019), popularized after the WikiCoref paper, can learn better representations of markables and surrounding sentences. Other than GAP, which targets a highly specific subtask, no study has investigated if contextualized embeddings encounter the same overfitting problem identified by Moosavi and Strube. Third, previous work may underestimate the performance degradation on WikiCoref in particular due to bias: In Moosavi and Strube (2018), embeddings were also trained on Wikipedia themselves, potentially making the model easier to learn coreference relations in Wikipedia text, despite limitations in other genres.

In this paper, we explore the generalizability of existing coreference models on a new benchmark dataset, which we make freely available. Compared with work using WikiCoref and GAP, our contributions can be summarized as follows:

- We propose OntoGUM, the largest open, gold dataset consistent with OntoNotes, with 168 documents (∼150K tokens, 19,378 mentions, 4,471 coref chains) in 12 genres,[1] including conversational genres, which complement OntoNotes for training and evaluation.
- We show that the SOTA neural model with contextualized embeddings encounter nearly 15% performance degradation on OntoGUM, showing that the overfitting problem is not overcome by contextualized language models.
- We give a genre-by-genre analysis for two popular systems, revealing relative strengths and weaknesses of current approaches and the range of easier/more difficult targets for coreference resolution.

[1] **Text:** News/Fiction/Bio/Academic/Forum/Travel/How-to/Textbook; **Speech:** Interview/Political/Vlog/Conversation.

2 Related Work

OntoNotes and similar corpora OntoNotes is a human-annotated corpus with documents annotated with multiple layers of linguistic information including syntax, propositions, named entities, word sense, and within document coreference (Weischedel et al., 2011; Pradhan et al., 2013). It covers three languages—English, Chinese and Arabic. The English subcorpus has 3,493 documents and ∼1.6 million words. WikiCoref, which is annotated for anaphoric relations, has 30 documents from English Wikipedia (Ghaddar and Langlais, 2016), containing 7,955 mentions in 1,785 chains, following OntoNotes guidelines.

GUM The Georgetown University Multilayer (GUM) corpus (Zeldes, 2017) is an open-source corpus of richly annotated texts from 12 types, including 168 documents and over 150K tokens. Though it originally contains more coreference phenomena than OntoNotes using more exhaustive guidelines, it also contains rich syntactic, semantic and discourse annotations which allow us to create the OntoGUM dataset described below. We also note that due to its smaller size (currently about 10% the size of the OntoNotes coreference dataset), it is not possible to train SOTA neural approaches directly on this dataset while maintaining strong performance.

Other corpora As mentioned above, GAP is a gender-balanced labeled corpus of ambiguous

pronoun-name pairs, used for out-of-domain evaluation but limited in coreferent types and genre. Several other comprehensive coreference datasets exist as well, such as ARRAU (Poesio et al., 2018) and PreCo (Chen et al., 2018), but these corpora cannot be used for out-of-domain evaluation because they do not follow the OntoNotes scheme. Their conversion has not been attempted to date.

Coreference resolution systems Prior to the introduction of deep learning systems, the coreference task was approached using deterministic linguistic rules (Lee et al., 2013; Recasens et al., 2013) and statistical approaches (Durrett and Klein, 2013, 2014). More recently, three neural models achieved SOTA performance on this task: 1) ranking the candidate mention pairs (Wiseman et al., 2015; Clark and Manning, 2016a), 2) modeling global features of entity clusters (Clark and Manning, 2015, 2016b; Wiseman et al., 2016), and 3) end-to-end (e2e) approaches with joint loss for mention detection and coreferent pair scoring (Lee et al., 2017, 2018; Fei et al., 2019). The e2e method has become the dominant one, gaining the best scores on OntoNotes. To investigate differences between deterministic and deep learning models on unseen data, we evaluate the two approaches on OntoGUM.

3 Dataset Conversion

GUM's annotation scheme subsumes all markables and coreference chains annotated in OntoNotes, meaning we do not need human annotation to recognize additional mentions in the conversion process, though mention boundaries differ subtly (e.g. for appositions and verbal mentions). Since GUM has gold syntax trees, we were able to process the entire conversion automatically. Additionally, most coreference evaluations use gold speaker information in OntoNotes, which is available in GUM (for *fiction*, *reddit* and spoken data) and could be assembled automatically as well.

The conversion is divided into two parts: removing coreference relations not included in the OntoNotes scheme, and removing or adjusting markables. For coreference relation deletion, we cut chains by removing expletive cataphora, and identifying the definiteness of nominal markables, since indefinites cannot be anaphors in OntoNotes. In addition to modifying existing mention clusters, we also remove particular coreference relations and mention spans, such as Noun-Noun compounding (only included in OntoNotes for proper-name modifiers), bridging anaphora, copula predicates, nested entities ('i-within-i'= single mentions containing coreferring pronouns), and singletons (all not included in OntoNotes). We note that singletons are removed as the final step, in order to catch singletons generated during the conversion process. We also contract verbal markable spans to their head verb, and merge appositive constructions into single mentions, following the OntoNotes guidelines. [2]

To evaluate conversion accuracy, three annotators, including an original OntoNotes project member, conducted an agreement study on 3 documents, containing 2,500 tokens and 371 output mentions. Re-annotating from scratch based on OntoNotes guidelines, the conversion achieves a span detection score of ∼96 and CoNLL coreference score of ∼92, approximately the same as human agreement scores on OntoNotes. After adjudication, the conversion was found to make only 8/371 errors, in addition to 2 errors due to mistakes in the original GUM data, meaning that degradation due to conversion errors is marginal, and consistency should be close to the variability in OntoNotes itself.

4 Experiments

We evaluate two systems on the 12 OntoGUM genres, using the official CoNLL-2012 scorer (Pradhan et al., 2012, 2014). The primary score is the average F1 of three metrics – MUC, B^3, and $CEAF_{\phi 4}$.

Deterministic coreference model We first run the deterministic system (dcoref, part of Stanford CoreNLP, Manning et al. 2014) on the OntoGUM benchmark, as it remains a popular option for off-the-shelf coreference resolution. As a rule-based system, it does not require training data, so we directly test it on OntoGUM's test set. However, POS tags, lemmas, and named-entity (NER) information are predicted by CoreNLP, which does have a domain bias favoring newswire. The system's multi-sieve structure and token-level features such as gender and number remain unchanged. We expect that the linguistic rules will function similarly across datasets and genres, notwithstanding biases of the tools providing input features to those rules.

SOTA neural model Combining the e2e approach with a contextualized LM and span masking is the current SOTA on OntoNotes. The system

[2]The code and dataset are publicly available at https://github.com/yilunzhu/ontogum.

Genre	MUC			B^3			$CEAF_{\phi4}$				Mention Detection		
	P	R	F1	P	R	F1	P	R	F1	Avg. F1	P	R	F1
	dcoref												
ac	35.1	37.5	36.2	32.6	34.4	33.5	35.7	37.5	36.6	35.4	48.3	51.3	49.8
bi	58.0	61.6	59.8	36.8	43.6	39.9	32.1	33.5	32.8	44.1	58.9	62.3	60.6
cn	62.2	52.9	57.1	40.5	36.7	38.5	37.1	38.2	37.6	44.4	76.6	67.8	72.0
fc	57.7	43.9	49.9	50.4	33.2	40.0	37.1	49.0	42.2	44.0	68.2	59.0	63.3
it	57.3	53.3	55.2	29.3	21.6	24.8	22.4	24.6	23.5	27.6	64.3	60.3	62.2
nw	57.6	55.2	56.4	45.7	42.3	44.0	39.6	32.5	35.7	45.3	44.0	50.2	46.9
rd	59.6	65.1	62.3	38.3	53.5	44.6	32.9	34.0	33.5	46.8	60.5	64.6	62.5
sp	50.6	56.2	53.2	40.1	43.9	41.9	46.5	38.6	42.2	45.8	63.5	64.2	63.9
tx	36.0	34.2	35.1	32.7	31.0	31.9	23.9	39.9	29.9	32.3	18.1	45.8	26.0
vl	63.6	69.4	66.4	56.4	60.8	58.5	31.4	36.2	33.6	52.8	76.4	76.8	76.6
vy	34.7	37.1	35.9	30.7	28.7	29.7	29.7	35.8	32.5	32.7	46.6	62.4	53.3
wh	35.8	24.2	28.9	30.0	24.5	27.0	29.9	34.0	31.8	29.2	50.0	42.9	46.2
All OntoGUM	45.7	47.0	46.3	17.1	38.1	37.6	33.4	37.3	35.3	39.7	56.2	59.1	57.6
OntoNotes	57.5	61.8	59.6	68.2	68.4	68.3	47.7	43.4	45.5	57.8	66.8	75.1	70.7
	Joshi et al. (2019a)												
ac	84.5	53.0	65.1	83.3	48.5	61.3	83.2	47.0	60.1	62.2	91.0	55.2	68.7
bi	85.8	74.7	79.8	61.4	64.3	62.8	65.4	49.9	56.6	66.4	87.7	74.5	80.5
cn	85.0	73.4	78.7	67.9	64.5	66.2	70.2	51.1	59.1	68.0	93.0	77.9	84.8
fc	87.0	62.5	73.0	78.8	54.1	64.1	62.5	53.1	57.4	64.8	91.1	67.7	77.7
it	83.9	71.8	77.4	76.1	60.4	67.3	72.9	50.6	59.7	68.2	85.9	70.4	77.3
nw	65.3	65.8	65.5	60.1	59.6	59.9	58.9	54.3	56.5	60.6	71.9	70.5	71.2
rd	76.7	67.4	71.7	67.5	60.3	63.7	69.5	40.5	51.1	61.7	85.3	68.1	75.8
sp	83.3	63.4	72.0	71.2	56.6	63.1	77.3	57.3	65.8	67.0	91.9	69.4	79.0
tx	50.0	66.6	57.1	45.2	65.7	53.6	55.6	55.6	55.6	55.5	60.0	72.2	65.5
vl	86.1	86.1	86.1	78.4	79.8	79.1	63.6	47.7	54.5	73.3	89.4	85.4	87.4
vy	69.0	70.4	69.7	52.7	64.1	57.9	65.9	53.0	58.8	62.1	78.9	75.5	77.2
wh	84.8	40.9	55.2	83.4	39.2	53.3	71.4	57.4	63.6	57.4	93.2	52.4	67.1
All OntoGUM	79.7	66.3	72.4	69.5	58.58	63.7	67.7	50.7	58.0	64.6	85.4	69.2	76.5
OntoNotes	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6	89.1	86.5	87.8

Table 2: Results on the OntoGUM's test dataset with the deterministic coref model (top) and the SOTA coreference model (bottom). The blue text is the lowest score across 12 genres and red text is the highest.

utilizes the pretrained SpanBERT-large model, fine-tuned on the OntoNotes training set. Hyperparameters are identical to the evaluation of OntoNotes test to ensure comparable results between the benchmarks. We note that while we choose the SOTA system as a 'best case scenario', most off-the-shelf neural NLP toolkits (e.g. spaCy) actually use somewhat simpler e2e models than SpanBERT-large, due to memory/performance constraints.

5 Results

OntoGUM vs. OntoNotes The last rows in each half of Table 2 give overall results for the systems on each benchmark. e2e+SpanBERT encounters a substantial degradation of 15 points (19%) on OntoGUM, likely due to lower test set lexical and stylistic overlap, including novel mention pairs. We note that its average score of 64.6 is somewhat optimistic, especially given that the system receives access to gold speaker information wherever available (including in *fc*, *cn* and *it*, some of the better scoring genres), which is usually unrealistic. dcoref, assumed to be more stable across genres, also sees losses on OntoGUM of over 18 points (30%). We believe at least part of the degradation may be due to mention detection, which is trained on different domains for both systems (see the last three columns in the table). These results suggest that input data from CoreNLP degrades substantially on OntoGUM, or that some types of coreferent expressions in OntoGUM are linguistically distinct from those in OntoNotes, or both, making OntoGUM a challenging benchmark for systems developed using OntoNotes.

Comparing genres Both systems degrade more on specific genres. For example, while *vl* (with gold speaker information) fares well for both systems, neither does well on *tx*, and even the SOTA system falls well below (or around) 60s for the *nw*, *wh* and *tx* genres. This might be surprising for *vl*, which contains transcripts of spontaneous unedited speech from YouTube Creative Commons vlogs quite unlike OntoNotes data; conversely the result is less expected for carefully edited texts which are somewhat similar to data in OntoNotes: OntoNotes

contains roughly 30% newswire text, and it is not immediately clear that GUM's *nw* section, which comes from recent Wikinews articles, differs much in genre. Examples (1)–(2) illustrate incorrectly predicted coreference chains from both sources and the type of language they contain.

(1) *I've been here just crushing ultrasounds ... I've been like crushing these all day today ... I got sick when I was on Croatia for vacation. I have no idea what it says, but I think they 're cough drops.* (example from a radiologist's vlog, incorrect: ultrasounds ≠ cough drops)

(2) *The report has prompted calls for all edible salt to be iodised ... Tasmania was excluded from the study - where a voluntary iodine fortification program using iodised salt in bread, is ongoing* (newswire example, incorrect span and coref: [the study - where a voluntary...])

These examples show that errors occur readily even in quite characteristic news writing, while genre disparity by itself does not guarantee low performance, as in the case of the vlogs whose lanugage is markedly different. In sum, these observations suggest that accurate coreference for downstream applications cannot be expected in some common well edited genres, despite the prevalence of news data in OntoNotes (albeit specifically from the Wall Street Journal, around 1990). This motivates the use of OntoGUM as a test set for future benchmarking, in order to give the NLP community a realistic idea of the range of performance we may see on contemporary data 'in the wild'.

We also suspect that prevalence of pronouns and gold speaker information produce better scores in the results. Table 3 ranks genres by their e2e CoNLL score, and gives the proportions of pronouns, as well as score rankings for span detection. Because pronouns are usually easier to detect and pair than nouns (Durrett and Klein, 2013), more pronouns usually means higher scores. On genres with more than 50% pronouns and gold speakers (*vl*, *it*, *cn*, *sp*, *fc*) e2e gets much higher results, while genres with few pronouns (<30%) have lower scores (*ac*, *vy*, *nw*). This diversity over 12 genres supports the usefulness of OntoGUM, which can evaluate the genrealizability of coreference systems.

	PRON (R)	Other (R)	Total	CoNLL	Span
vl	600 (.66)	309 (.34)	909	1	1
it	1223 (.45)	1485 (.55)	2708	2	6
cn	729 (.61)	323 (.39)	1052	3	2
sp	245 (.40)	364 (.60)	609	4	4
bi	796 (.34)	1529 (.66)	2325	5	3
fc	1700 (.61)	1091 (.39)	2791	6	5
ac	262 (.21)	997 (.79)	1259	7	10
vy	300 (.22)	1053 (.78)	1353	8	7
rd	1337 (.55)	1077 (.45)	2414	9	8
nw	340 (.19)	1483 (.81)	1823	10	9
wh	1001 (.47)	1129 (.53)	2130	11	11
tx	165 (.34)	315 (.66)	480	12	12

Table 3: Mention-type counts (ratios) & ranks of SOTA scores by genre (CoNLL score + span detection).

6 Conclusion

This paper presented OntoGUM, the largest open, gold coreference dataset following the OntoNotes scheme, adding several new genres (including more spoken data) to the OntoNotes family. The corpus is automatically converted from GUM by modifying the existing markable spans and coreference relations using multi-layer annotations, such as dependency trees. Results showed a lack of generalizability of existing systems, especially in genres low in pronouns and lacking speaker information. We suspect that at least part of the success of SOTA approaches is due to correct mention detection and high matching scores in genres rich in pronouns, and more so with gold speaker information. Success for other types of mentions in OntoNotes data appears to be much more sensitive to lexical features, performing well on the benchmark test set with high lexical overlap to the training data, but degrading very substantially outside of it, even on newswire texts from our OntoGUM data. This supports use of this challenging dataset for future work, which we hope will benefit evaluations of systems targeting the OntoNotes standard.

References

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of EMNLP 2018*, pages 172–181, Brussels.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of ACL-IJCNLP 2015, Long Papers*, pages 1405–1415, Beijing, China.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking

coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of ACL 2016, Long Papers*, pages 643–653, Berlin.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL 2019*, pages 4171–4186, Minneapolis, MN.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of EMNLP 2013*, pages 1971–1982, Seattle, WA.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *TACL*, 2:477–490.

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of ACL 2019*, pages 660–665, Florence, Italy.

Abbas Ghaddar and Phillippe Langlais. 2016. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of LREC 2016)*, pages 136–142, Portorož, Slovenia.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019a. SpanBERT: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Mandar Joshi, Omer Levy, Daniel S. Weld, and Luke Zettlemoyer. 2019b. BERT for coreference resolution: Baselines and analysis. In *Proceedings of EMNLP-IJCNLP 2019*, pages 5803–5808, Hong Kong, China.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of EMNLP 2017*, pages 188–197, Copenhagen, Denmark.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of NAACL 2018, Short Papers*, pages 687–692, New Orleans, LA.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL 2014 System Demonstrations*, pages 55–60.

Nafise Sadat Moosavi and Michael Strube. 2017. Lexical features in coreference resolution: To be used with caution. In *Proceedings of ACL 2017, Short Papers*, pages 14–19, Vancouver, Canada.

Nafise Sadat Moosavi and Michael Strube. 2018. Using linguistic features to improve the generalization capability of neural coreference resolvers. In *Proceedings of EMNLP 2018*, pages 193–203, Brussels.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora resolution with the ARRAU corpus. In *Proceedings of CRAC 2018*, pages 11–22, New Orleans, LA.

Massimo Poesio, Roland Stuckardt, and Yannick Versley. 2016. *Anaphora resolution*. Springer.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of ACL 2014*, pages 30–35, Baltimore, MD.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of CoNLL 2013*, pages 143–152, Sofia.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of NAACL 2013*, pages 627–633, Atlanta, Georgia.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *TACL*, pages 605–617.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. OntoNotes: A large training corpus for enhanced processing. In Joseph Olive, Caitlin Christianson, and John McCary, editors, *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of ACL-IJCNLP 2015, Long Papers*, pages 1416–1426, Beijing, China.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of NAACL 2016*, pages 994–1004, San Diego, CA.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

In Factuality: Efficient Integration of Relevant Facts for Visual Question Answering

Peter Vickers[*] and **Nikolaos Aletras**[*] and **Emilio Monti**[+] and **Loïc Barrault**[*]

[*]Department of Computer Science
University of Sheffield
{pgjvickers1, n.aletras, l.barrault}
@sheffield.ac.uk

[+]Amazon United Kingdom
monti@amazon.co.uk

Abstract

Visual Question Answering (VQA) methods aim at leveraging visual input to answer questions that may require complex reasoning over entities. Current models are trained on labelled data that may be insufficient to learn complex knowledge representations. In this paper, we propose a new method to enhance the reasoning capabilities of a multi-modal pretrained model (Vision+Language BERT) by integrating facts extracted from an external knowledge base. Evaluation on the KVQA dataset benchmark demonstrates that our method outperforms competitive baselines by 19%, achieving new state-of-the-art results. We also perform an extensive analysis highlighting the limitations of our best performing model through an ablation study.

1 Introduction

Visual Question Answering (VQA) is a popular multi-modal task of answering a question about an image. It tracks both inter-modal interactions and reasoning capabilities of models (Wang et al., 2017; Marino et al., 2019). Recent studies have tested compositional reasoning (Johnson et al., 2016; Hudson and Manning, 2019) and the integration of external knowledge (Wang et al., 2017, 2016; Shah et al., 2019; Marino et al., 2019) for VQA. In this paper, we address Knowledge-aware VQA (KVQA) (Shah et al., 2019)[1] , defined as a VQA task where it is not reasonable to expect a model without access to a knowledge base to be able to answer the questions in the test set.

In a uni-modal textual context, both synthetic dataset (Kassner et al., 2020) and task-driven (Ding et al., 2020) studies of neural models have shown significant competence at symbolic reasoning. This is encouraging, as neural pretrained Language Models such as BERT (Devlin et al., 2019) achieve state-of-the-art results in a wide range of natural language inference tasks and benchmarks such as Natural Language Inference (Bowman et al., 2015). (Rajani et al., 2019) uses pretraining on a domain-specific dataset to improve CommonsenseQA by 10% absolute accuracy. Tamborrino et al. (2020) develop an improved training objective to improve COPA by 10% absolute accuracy.

Bouraoui et al. (2020) find that BERT is capable of relational induction, whilst Broscheit (2019); Petroni et al. (2020) find that BERT stores non-trivial world-knowledge.

Previous work has argued that restriction to a uni-modal context may itself impair reasoning performance (Barsalou, 2008; Li et al., 2020). In a bi-modal Vision + Language (V+L) context, datasets such as CLEVR and GQA allow for the evaluation of both model reasoning and language grounding. Within this setting, Ding et al. (2020) and Lu et al. (2020) show that appropriate neural models trained on large quantities of data can exhibit accurate reasoning.

In this paper, we propose a new method of applying a massively pretrained V+L BERT model (Chen et al., 2020) to the KVQA task (Shah et al., 2019). Our method is able to learn a set of reasoning types (confirming findings in Ding et al. (2020)) but can increase performance even more by incorporating external factual information. KVQA answers require attending to a knowledge base, allowing us to quantify the contribution of both explicit and implicit knowledge extracted from supervised training data. We also quantify the degree to which corpus bias makes certain question types harder, and outline how future datasets may be better balanced.

Our contributions are as follows:

- We perform factual integration into a V+L BERT-based model architecture VQA, leading to 19.1% accuracy improvement over previous baselines on KVQA.

[1]For data, examples, and licence information, please see https://malllabiisc.github.io/resources/kvqa/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 468–475
August 1–6, 2021. ©2021 Association for Computational Linguistics

- We evaluate our model's reasoning capabilities through an ablation study, proposing explanations for poor performance on certain question types as well as highlighting our model's strong preference for text and facts over the image modality.

- We conduct a bias study of the KVQA dataset, revealing both strengths and potential improvements for future VQA datasets.

2 Related Work

VQA tasks explicitly encourage grounded reasoning (Antol et al., 2015), with emphasis on a variety of sub-domains, such as commonsense (Zellers et al., 2019), compositionality and grounding (Suhr et al., 2020), factual reasoning (Wang et al., 2017) or external knowledge reasoning (Wang et al., 2016; Marino et al., 2019; Shah et al., 2019).

State-of-the-art systems for external knowledge VQA are based on Memory networks (MemNet, (Weston et al., 2014)). In Shah et al. (2019), the facts are extracted from the Knowledge Graph (KG) by considering the visual (from image) and eventually textual (from Wikipedia caption) entities. They are then embedded using a Bi-LSTM encoder and fed into the memory. After the question is embedded in a similar way, the resulting representation is used to query the memory by soft attention. Several stacked memory layers are used to better model multi-hop facts.

Wang et al. (2016, 2017) introduce two datasets, KB-VQA and FVQA respectively, and address the task with systems that perform searches in a visual knowledge graph formed from the image and a KB. The question is first mapped to a query of the form ⟨visual object, relationship, answer source⟩, which is then used to extract the supporting facts from the KB. They report improved results when compared to systems using LSTM, SVM and hierarchical co-attention (Lu et al., 2016).

In Marino et al. (2019), the OK-VQA is presented with some baseline results obtained with MUTAN (Ben-younes et al., 2017), a multimodal tensor-based Tucker decomposition which models interactions between visual (from CNN) and textual (from RNN) representations. Those systems exhibit rather low performance compared to those obtained on standard VQA, demonstrating that the corpus requires external knowledge to be solved correctly.

Recent work has introduced methods to incorporate visual information to create Vision+Language BERT models through joint multimodal embeddings (Chen et al., 2020; Su et al., 2019; Lu et al., 2019). First, image and text are embedded into the same space, and then Transformer networks are applied as in the standard BERT model (Devlin et al., 2019).

Our work is most similar to that of Shah et al. (2019) since the same preprocessing pipeline is used. However, our system does not use a memory network, and instead relies on on a BERT-based model (UNITER, see section 3) to model the relationship between question, facts, and image with self-attention layers.

3 Methodology

To answer KVQA with Neural models, we first take the V+L BERT model UNITER (Chen et al., 2020) with the highest score on the commonsense VQA task, VCR (Zellers et al., 2019).

In order to allow UNITER to accept external KG facts, we cast these facts to a textual form 'Entity_1 Relation Entity_2'. To keep the input facts count small, we perform a *conditional search* of the KG. The KVQA task consists in finding a^*:

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, p(a|q, i, K) \approx \underset{a \in A}{\operatorname{argmax}}\, p(a|q, i, k_{i,q}) \quad (1)$$

where a^* is the correct answer out of candidate set A; and q, i, and K are a question, image and knowledge base, respectively. As shown, we may reduce the KG through a conditional search to find the relevant subset of facts $k_{i,q}$.

To define the subset $k_{i,q}$, we follow Shah et al. (2019) in extracting all facts from the knowledge base that are up to two hops from any entities detected by the textual entity linking or the face detection.

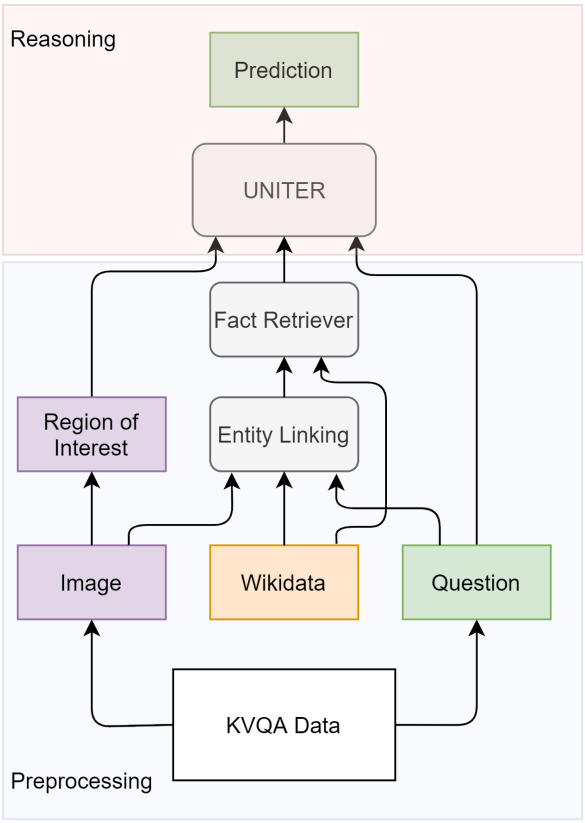


Figure 1: Our Model

Our model, as presented in section 2 consists of two stages: preprocessing, which implements relevant fact extraction, and reasoning, which selects an answer from the question, facts, and image features.

3.1 Preprocessing Stage

For preprocessing and fact acquisition, we broadly reproduce the fact and feature extraction process used in Shah et al. (2019). We perform object detection with the Faster R-CNN network (Ren et al., 2017). A seven-dimensional normalised size and location vector is concatenated with the Faster R-CNN features.

For person detection, we use MTCNN (Zhang et al., 2016) and Facenet (Schroff et al., 2015) models, pretrained on the MS-celeb-1M (Guo et al., 2016) dataset, to generate 128-dimensional embeddings. We predict names by nearest-neighbour comparison with the KVQA reference dataset. We treat the name identification as a multi-class classification problem, achieving a Micro-F1 of 0.539. Since this is lower than reported in Shah et al. (2019), we follow them in applying a textual entity linker (van Hulst et al., 2020) over supplied image descriptions. This setup achieves a per-image Micro-F1 of 0.686.

Normalised image location facts are generated from these detections, such as 'Barack Obama at 42 78', which would indicate that the centre bounding box for Barack Obama is at normalised (0-100) position x=42, y=78 of the image. We use the names of identified entities to query Shah et al.'s 2019 reduced Wikidata graph (Vrandečić and Krötzsch, 2014) up to two hops. The extracted facts are finally cast to the form 'subject relation object'.

3.2 Reasoning Stage

The neural model we use, UNITER, is pretrained on MS COCO (Lin et al., 2014), Visual Genome (Krishna et al., 2016), Conceptual Captions (Sharma et al., 2018), and SBU Captions (Ordonez et al., 2011). It is a multi-task system that is trained on performing Masked Language Modeling, Image-Text Matching, and Masked Region Modeling (Chen et al., 2020).

4 Experimental Setup

We select the KVQA dataset for two reasons: to our knowledge, it is the largest external knowledge dataset (with 183k questions), and the questions are annotated with their reasoning types. We use accuracy as the evaluation metric and provide results over both the entire dataset and also for each question type as provided in the KVQA dataset.

The baseline systems for KVQA are those presented in (Shah et al., 2019) and discussed in section 2. The first baseline is a stacked BLSTM encoder, operating over question and facts. This system has an overall accuracy of 48.0% . The second is the MemNet architecture and has the previously highest performing baseline accuracy at 50.2%.

We use the UNITER_BASE pretrained model available at the ChenRocks GitHub repository[2] with custom classification layers (MLP +softmax output layer). For task training, we merge retrieved facts with the question, dividing each statement with the '[SEP]' token, following research that indicates that this token induces partitioning and pipelining of information across attention layers (Clark et al., 2019). The textual input stream is tokenised with the HuggingFace 'bert-base-uncased' tokeniser (Wolf et al., 2020). We set the maximum WordPiece sequences length to 412, the maximum visual objects count to 100, the learning rate to

[2] https://github.com/ChenRocks/UNITER

Question Type	Model		Entropy (Base 2)
	MemNet	**UNITER**	
1-Hop	61.0	**65.7**	7.8
1-Hop Counting	-	**78.0**	1.4
1-Hop Subtraction	-	**28.6**	4.3
Boolean	75.1	**94.6**	1.1
Comparison	50.5	**90.4**	2.1
Counting	49.5	**79.4**	2.3
Intersection	72.5	**79.4**	1.2
Multi-Entity	43.5	**77.1**	3.3
Multi-Hop	53.2	**87.9**	3.7
Multi-Relation	45.2	**75.2**	7.1
Spatial	**48.1**	21.2	11.5
Subtraction	**40.5**	34.4	6.0
Overall	50.2	69.3	7.6

Table 1: Results in terms of % accuracy of the considered systems break down into question types along with the question types distribution (last column).

Question Type	Q+F+I	Q+F	Q+I	F+I	Q	F	I
1-Hop	65.7	65.7	32.4	3.9	32.4	3.8	4.5
1-Hop Counting	78.0	78.0	30.3	0.0	30.3	0.0	0.0
1-Hop Subtraction	28.9	28.6	28.8	0.8	30.3	0.6	6.5
Boolean	94.6	94.6	55.2	1.3	55.2	1.0	10.5
Comparison	90.4	90.4	38.7	1.0	38.7	0.9	10.7
Counting	79.4	79.4	66.1	0.6	65.9	0.4	1.4
Intersection	79.4	79.4	61.0	0.4	60.6	0.3	0.0
Multi-Entity	77.1	77.1	41.3	0.8	41.2	0.7	6.4
Multi-Hop	87.9	87.9	29.0	0.8	28.9	0.8	0.0
Multi-Relation	75.2	75.2	25.1	3.0	25.0	3.0	2.5
Spatial	21.2	21.2	0.0	13.0	0.0	13.0	0.0
Subtraction	34.4	34.4	1.3	1.0	0.9	0.7	0.0
Overall	69.3	69.3	31.6	3.1	31.5	3.0	3.6

Table 2: Ablation Study of Information. Q=Question, I=Image, F=Facts. Image refers to the Image feature stream. Results are expressed as % accuracy by question type.

8×10^{-5} and use AdamW (Loshchilov and Hutter, 2017) as optimizer. Once preprocessing is completed, we train the UNITER model with the cross-entropy objective function for 80,000 iterations, which we empirically found to guarantee convergence.

5 Results

Table 1 shows the results of our system (UNITER), using a question label break-down similar to Shah et al. (2019). Overall, we observe that our system outperforms the previous baseline MemNet setting (see 'World+WikiCap+ORG' in Shah et al. (2019)) with an absolute improvement of 19%.

Our results show that UNITER is learning to perform reasoning more accurately than MemNet in all but two cases. In the question types involving multiple entities ('Multi-Entity', 'Multi-Hop', 'Multi-Relation'), the increase is the greatest, suggesting that UNITER is able to robustly learn these reasoning here. We speculate that stacked self-attention layers in BERT are able to better attend to the many involved entities than MemNet.

We now discuss the performance of our model on its weakest categories, namely 'Subtraction' and 'Spatial'. The poor performance on 'Subtraction' questions confirms previous results that BERT-like models require specialised pretraining for numerical reasoning tasks (Geva et al., 2020). In the case of our model specifically, we note the lack of numerical reasoning tasks in UNITER's pretraining regime. 'Spatial' is the model's least accurate question type (21.4%) and the biggest absolute decrease from MemNet (-26.7%). This question type requires two-hop reasoning where the second hop is a numerical operation of the form $\underset{y}{\text{argmin}}(x_i - y_i)$. Both of these have been shown to be problematic for BERT (Kassner et al., 2020; Geva et al., 2020).

6 Analysis

UNITER performs well at the reasoning tasks in general, with the most surprising result being that it apparently does better at multi-hop reasoning than one-hop. We believe that this can be explained by the presence of unbalanced distribution of answer types in the dataset perturbing the results (see Table 1). We discuss this in Section 6.1.

In order to better understand the reasoning capability of our model and the impact of each input modality, we perform an inference time ablation study, presented in Table 2.

Ablation of Image features (column 'Q+F') does not change the performance, suggesting that the model is not attending to image features. To confirm this hypothesis, we performed an experiment with adversarial images, obtaining very similar results for each question type and the same overall score (69.30%). We explain this behaviour by the fact that the preprocessing pipeline extracts all the required information as explicit facts which the model prefers over the more ambiguous visual features. We leave a deeper analysis for further work.

An interesting case is the 'Spatial' questions, where facts alone are able to correctly answer 13% of the questions. This is likely the result of the answers to this question type being entities present in the facts. Again, we observe that the model is not able to learn this information from the visual features.

	Train Ablation		Adversarial Modality*	
Question Type	Q+I	Q	I	F
1-Hop	47.09	38.5	65.9	31.3
1-Hop Counting	66.1	61.5	75.2	50.5
1-Hop Subtraction	29.4	29.7	28.1	26.2
Boolean	83.9	67.3	94.1	57.5
Comparison	83.4	60.3	90.6	47.8
Counting	75.4	75.2	78.9	70.2
Intersection	67.6	67.9	76.8	61.2
Multi-Entity	69.4	57.2	76.4	47.6
Multi-Hop	56.5	50.2	87.9	38.4
Multi-Relation	47.3	38.9	75.2	28.3
Spatial	3.3	1.2	21.1	0.0
Subtraction	2.1	2.6	39.2	1.6
Overall	47.0	40.8	69.3	32.8

Table 3: Further Ablation and Adversarial Studies. *Adversarial Modality indicates that the sample from that modality was randomly assigned from the entire data split

6.1 Bias Studies

We briefly discuss the corpus bias, a well-known concern in VQA (Goyal et al., 2019). We consider question difficulty across three parameters: reasoning difficulty, task design, and corpus bias. Certain question types are inherently more complex, as discussed in Section 5. Additionally, the task may have different numbers of answer classes per task, effectively weakening any priors models might form (see Entropy column in Table 1). Finally, an unbalanced dataset may cause certain reasoning types to be underrepresented, making it harder for models to learn for them. 'Spatial' and 'Substraction' questions are among the least represented in the training dataset, which increase their difficulty for the model.

Unseen answer classes are also an issue. For 'Spatial' questions, only 54.2% of the test answers (output classes) are actually seen during training, placing an upper bound on accuracy. We find 98.4% of 'Spatial' questions the model answered correctly and 95.7% of 'Spatial' question the model answered incorrectly were supplied with adequate facts by the preprocessing pipeline.

Training time ablation and adversarial experiments To further probe the task, we perform a training time ablation with first facts, and then facts and images removed (see Table 3). In this we seek to exhibit the capability of our model to leverage the available modalities and to compensate for the missing ones.

Through comparing the training time and inference time ablations, we can better understand the importance of a modality to solving the task.

Through comparing train and inference ablation of facts ('Q+I' column of Table 3 and of Table 2) we observe that when facts are unavailable at train time, the model attends to images to obtain 47.0% accuracy, which is 15.4% more than the 31.6% obtained by the corresponding inference time ablation. This indicates that the visual modality can provide useful information for this task.

We observe a similar trend in the fact and image ablation setting ('Q' column of Table 3 and of Table 2) that the model is able to greater leverage questions to make accurate predictions when additional modalities are never available.

We also perform adversarial checks, where random images or facts from the data split are presented at inference time. These align closely with the ablation study, with adversarial images (Column 'I' of Table 3) performing within 0.1% of blanked images (Column 'Q+F' of Table 3) and adversarial facts (Column 'F' of Table 3) performing within 1% of blanked facts (Column 'Q+I' of Table 3). These results confirm the importance of factual data and the unimportance of raw image features to a model trained on the full data.

7 Conclusion and Future Work

We evaluated our model and found that it improves on the previous state of the art by a substantial margin (19.1%). An ablation study revealed the specific strengths and weaknesses of our model on certain question categories when evaluated on the KVQA dataset. We show that the UNITER model is not actually using the visual input.

In the future, we seek to create a large external knowledge dataset designed following KVQA with more entities besides persons to encourage grounded reasoning, and better calibration of answer types. We will also consider pretraining our model on closely related tasks. This will help to form a model capable of learning robust reasoning with a high degree of spatial specificity and entity discrimination.

Acknowledgements

Peter Vickers is supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by the UK Research and Innovation grant EP/S023062/1.

Ethical Statement

This work is based on the open-source KVQA dataset, an English multimodal dataset, and the Wikidata knowledge base (also in English). No English-specific preprocessing was used for this research and the UNITER model is language agnostic, which tends to suggest that this could generalize to other languages. We will make our code publicly available to ensure the reproducibility of our experiments in the following repository

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2015 Inter, pages 2425–2433.

Lawrence W Barsalou. 2008. Grounded cognition. *Annual Review of Psychology*, 59:617–645.

Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2631–2639.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing Relational Knowledge from BERT. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7456–7463.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics (ACL).

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*, pages 677–685. Association for Computational Linguistics.

Yen Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12375 LNCS:104–120.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186.

David Ding, Felix Hill, Adam Santoro, and Matt Botvinick. 2020. Object-based attention for spatio-temporal reasoning: Outperforming neuro-symbolic models with flexible distributed architectures.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting Numerical Reasoning Skills into Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2019. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision*, 127(4):398–414.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. 2016. MS-celeb-1M: A dataset and benchmark for large-scale face recognition. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9907 LNCS, pages 87–102.

Drew A Hudson and Christopher D Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 6693–6702.

Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: An Entity Linker Standing on the Shoulders of Giants. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2016. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:1988–1997.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are Pretrained Language Models Symbolic

Reasoners over Knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Ranjay Krishna, Justin Johnson, Yannis Kalantidis Yahoo, David Ayman Shamma, Yuke Zhu, Oliver Groth, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations Human trajectory forecasting View project hybrid intrusion detction systems View project Visual Genome Connecting Language and Vision Using Crowdsourced Dense Image A. *Article in International Journal of Computer Vision*, 123(1):32–73.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020. What Does BERT with Vision Look At? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275, Online. Association for Computational Linguistics.

Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, PART 5, pages 740–755. Springer Verlag.

Ilya Loshchilov and Frank Hutter. 2017. Fixing Weight Decay Regularization in Adam. *CoRR*, abs/1711.0.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Advances in Neural Information Processing Systems*, volume 32, pages 13–23. Curran Associates, Inc.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-Task Vision and Language Representation Learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems*, volume 29, pages 289–297. Curran Associates, Inc.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:3190–3199.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Advances in Neural Information Processing Systems*, volume 24, pages 1143–1151. Curran Associates, Inc.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. Language models as knowledge bases? In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 2463–2473. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 4932–4942.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 815–823.

Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. 2019. KVQA: Knowledge-Aware Visual Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8876–8884.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 2556–2565. Association for Computational Linguistics (ACL).

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. *arXiv*.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2020. A corpus for reasoning about natural language grounded in photographs. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 6418–6428. Association for Computational Linguistics (ACL).

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training Is (Almost) All You Need: An Application to Commonsense Reasoning. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 3878–3887, Online. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. 2017. Explicit knowledge-based reasoning for visual question answering. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 0, pages 1290–1296. International Joint Conferences on Artificial Intelligence.

Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. 2016. FVQA: Fact-based Visual Question Answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2413–2427.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 6713–6724.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.

Zero-shot Fact Verification by Claim Generation

Liangming Pan[1,2] **Wenhu Chen**[3] **Wenhan Xiong**[3]
Min-Yen Kan[2] **William Yang Wang**[3]
[1]NUS Graduate School for Integrative Sciences and Engineering
[2]School of Computing, National University of Singapore, Singapore
[3]University of California, Santa Barbara, CA, USA
liangmingpan@u.nus.edu
{wenhuchen, xwhan, william}@cs.ucsb.edu
kanmy@comp.nus.edu.sg

Abstract

Neural models for automated fact verification have achieved promising results thanks to the availability of large, human-annotated datasets. However, for each new domain that requires fact verification, creating a dataset by manually writing claims and linking them to their supporting evidence is expensive. We develop QACG, a framework for training a robust fact verification model by using automatically-generated claims that can be supported, refuted, or unverifiable from evidence from Wikipedia. QACG generates question–answer pairs from the evidence and then convert them into different types of claims. Experiments on the FEVER dataset show that our QACG framework significantly reduces the demand for human-annotated training data. In a zero-shot scenario, QACG improves a RoBERTa model's F_1 from 50% to 77%, equivalent in performance to 2K+ manually-curated examples. Our QACG code is publicly available.[1]

1 Introduction

Fact verification aims to validate a claim in the context of evidence. This task has attracted growing interest with the rise in disinformation in news and social media. Rapid progress has been made by training large neural models (Zhou et al., 2019; Liu et al., 2020b; Zhong et al., 2020) on the FEVER dataset (Thorne et al., 2018), containing more than 100K human-crafted (evidence, claim) pairs based on Wikipedia.

Fact verification is demanded in many domains, including news articles, social media, and scientific documents. However, it is not realistic to assume that large-scale training data is available for every new domain that requires fact verification. Creating training data by asking humans to write claims and search for evidence to support/refute them can be extremely costly.

We address this problem by exploring the possibility of automatically *generating* large-scale (evidence, claim) pairs to train the fact verification model. We propose a simple yet general framework **Q**uestion **A**nswering for **C**laim **G**eneration (QACG) to generate three types of claims from any given evidence: 1) claims that are `supported` by the evidence, 2) claims that are `refuted` by the evidence, and 3) claims that the evidence does Not have Enough Information (`NEI`) to verify.

To generate claims, we utilize *Question Generation (QG)* (Zhao et al., 2018; Liu et al., 2020a; Pan et al., 2020), which aims to automatically ask questions from textual inputs. QG has been shown to benefit various NLP tasks, such as enriching QA corpora (Alberti et al., 2019), checking factual consistency for summarization (Wang et al., 2020), and data augmentation for semantic parsing (Guo et al., 2018). To the best of our knowledge, we are the first to employ QG for fact verification.

As illustrated in Figure 1, given a passage P as the evidence, we first employ a *Question Generator* to generate a question–answer pair (Q, A) for the evidence. We then convert (Q, A) into a claim C (*QA-to-Claim*) based on the following logical assumptions: a) if P can answer Q and A is the correct answer, then C is a `supported` claim; b) if P can answer Q but A is an incorrect answer, then C is a `refuted` claim; c) if P cannot answer Q, then C is a `NEI` claim. The Question Generator and the QA-to-Claim model are off-the-shelf BART models (Lewis et al., 2020), finetuned on SQuAD (Rajpurkar et al., 2016) and QA2D (Demszky et al., 2018) datasets.

We generate 100K (evidence, claim) pairs for each type of claim, which we then use to train a RoBERTa (Liu et al., 2019) model for fact verification. We evaluate the model on three test sets

[1] https://github.com/teacherpeterpan/Zero-shot-Fact-Verification

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 476–483
August 1–6, 2021. ©2021 Association for Computational Linguistics

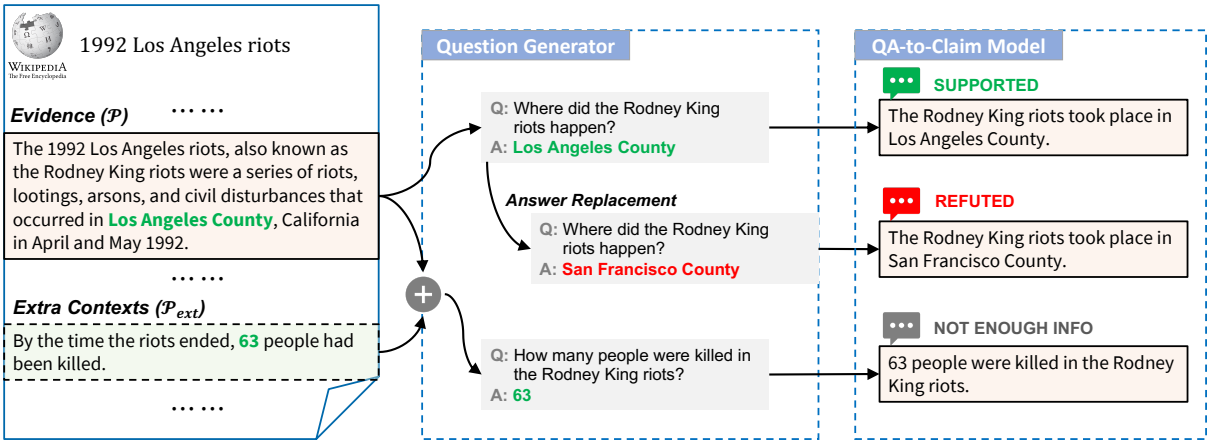


Figure 1: Overview of our QACG framework, consisting of two modules: 1) *Question Generator* generates questions from the evidence $\mathcal{P}$ and the extra contexts $\mathcal{P}_{ext}$ given different answers extracted from the passage (in green), and 2) *QA-to-Claim* converts question-answer pairs into claims with different labels.

based on the FEVER dataset. Although we do not use any human-labeled training examples, the model achieves over 70% of the F_1 performance of a fully-supervised setting. By finetuning the model with only 100 labeled examples, we further close the performance gap, achieving 89.1% of fully-supervised performance. The above results show that pretraining the fact verification model with generated claims greatly reduces the demand for in-domain human annotation. When evaluating the model on an unbiased test set for FEVER, we find that training with generated claims also produces a more *robust* fact verification model.

In summary, our contributions are:

- To the best of our knowledge, this is the first work to investigate zero-shot fact verification.
- We propose QACG, a novel framework to generate high-quality claims via question generation.
- We show that the generated training data can greatly benefit the fact verification system in both zero-shot and few-shot learning settings.

2 Methodology

Given a claim $\mathcal{C}$ and a piece of evidence $\mathcal{P}$ as inputs, a *fact verification* model $\mathcal{F}$ predicts a label $\mathcal{Y} \in \{\texttt{supported}, \texttt{refuted}, \texttt{NEI}\}$ to verify whether $\mathcal{C}$ is supported, refuted, or can not be verified by the information in $\mathcal{P}$.

For the *zero-shot* setting, we assume no human-annotated training example is available. Instead, we generate a synthetic training set based on our QACG framework to train the model.

2.1 Question Generator and QA-to-Claim

As illustrated in Figure 1, our claim generation model QACG has two major components: a *Question Generator* $\mathcal{G}$, and a *QA-to-Claim* model $\mathcal{M}$.

The **Question Generator** takes as input an evidence $\mathcal{P}$ and a text span A from the given evidence and aims to generate a question Q with A as the answer. We implement this with the BART model (Lewis et al., 2020), a large transformer-based sequence-to-sequence model pretrained on 160GB of text. The model is finetuned on the SQuAD dataset processed by Zhou et al. (2017), where the model encodes the concatenation of the SQuAD passage and the answer text and then learns to decode the question. We evaluate the question generator using automatic and human evaluation and investigate its impact on fact verification in Appendix A.

The **QA-to-Claim Model** takes as inputs Q and A, and outputs the declarative sentence C for the (Q, A) pair, as shown in Figure 1. We also treat this as a sequence-to-sequence problem and finetune the BART (Lewis et al., 2020) model on the QA2D dataset (Demszky et al., 2018), which contains the human-annotated declarative sentence for each (Q, A) pair in SQuAD.

2.2 Claim Generation

Given the pretrained question generator $\mathcal{G}$ and the QA-to-Claim model $\mathcal{M}$, we then formally introduce how we generate claims with different labels.

`Supported` claim generation. Given an evidence P, we use named entity recognition to identify all entities within P, denoted as $\mathcal{E}$. For each

entity $a \in \mathcal{E}$, we treat each a in turn as an answer and generate a question $q = \mathcal{G}(\mathcal{P}, a)$ with the question generator. The question–answer pair (q, a) are then sent to the QA-to-Claim model to generate the `supported` claim $c = \mathcal{M}(q, a)$.

`Refuted` claim generation. To generate a `refuted` claim, after we generate the question–answer pair (q, a), we use *answer replacement* (shown in Figure 1) to replace the answer a with another entity a' with the same type such that a' becomes an incorrect answer to the question q. Using a as the query, we randomly sample a phrase from the top-5 most similar phrases in the pretrained Sense2Vec (Trask et al., 2015) as the replacing answer a'. The new pair (q, a') is then fed to the QA-to-Claim model to generate the `refuted` claim.

To avoid the case that a' is still the correct answer, we define rules to ensure that the a' has less lexical overlap with a. However, this problem is sometimes non-trivial and cannot be completely avoided. For example, for the QA pair: ("Who is the producer of Avatar?"; "James Cameron"), another valid answer a' is "Jon Landau", who happens to be another producer of Avatar. However, we observe that such coincidences rarely happen: among the 100 randomly sampled claims, we only observed 2 such cases. Therefore, we leave them as the natural noise of the generation model.

`NEI` claim generation. We need to generate a question q' which is relevant but cannot be answered by $\mathcal{P}$. To this end, we link $\mathcal{P}$ back to its original Wikipedia article $\mathcal{W}$ and expand the evidence with additional contexts $\mathcal{P}_{ext}$, which are five randomly-retrieved sentences from $\mathcal{W}$ that are not present in $\mathcal{P}$. In our example in Figure 1, one additional context retrieved is "By the time the riots ended, 63 people had been killed". We then concatenate $\mathcal{P}$ and $\mathcal{P}_{ext}$ as the expanded evidence, based on which we generate a `supported` claim given an entity in $\mathcal{P}_{ext}$ as the answer (*e.g.*, "63"). This results in a claim relevant to but unverifiable by the original evidence $\mathcal{P}$.

3 Experiments

By applying our QACG model to each of the $18,541$ Wikipedia articles in the FEVER training set, we generate a total number of $176,370$ `supported` claims, $360,924$ `refuted` claims, and $258,452$ `NEI` claims. Our generated data is around five times the size of the human-annotated claims in FEVER. We name this generated dataset as QACG-*Full*. We then create a balanced dataset QACG-*Filtered* by randomly sampling $100,000$ samples for each class. Statistics of the FEVER and the generated dataset are in Appendix B.

Evaluation Datasets. We evaluate fact verification on three different test sets based on FEVER: **1) FEVER-S/R**: Since only the `supported` and `refuted` claims are labeled with gold evidence in FEVER, we take the claim–evidence pairs of these two classes from the FEVER test set for evaluation. **2) FEVER-Symmetric**: this is a carefully-designed unbiased test set designed by Schuster et al. (2019) to detect the robustness of the fact verification model. Note that only `supported` and `refuted` claims are present in this test set. **3) FEVER-S/R/N**: The full FEVER test set are used for a three-class verification. We follow Atanasova et al. (2020) to use the system of Malon (2019) to retrieve evidence sentences for `NEI` claims.

Fact Verification Models. As shown in Table 1, we take a BERT model (S1) and a RoBERTa model (S2) fine-tuned on the FEVER training set as the *supervised* models. Their corresponding *zero-shot* settings are Rows U5 and U6, where the models are trained on our generated QACG-*Filtered* dataset. Note that for binary classification (FEVER-S/R and FEVER-Symmetric), only the `supported` and `refuted` claims are used for training, while for FEVER-S/R/N, the full training set is used.

We employ four baselines that also do not need any human-annotated claims to compare with our method. *Random Guess* (U1) is a weak baseline that randomly predicts the class label. *GPT2 Perplexity* (U2) predicts the class label based on the perplexity of the claim under a pretrained GPT2 (Radford et al., 2019) language model, following the assumption that "misinformation has high perplexity" (Lee et al., 2020a). *MNLI-Transfer* (U3) trains a BERT model for natural language inference on the MultiNLI corpus (Williams et al., 2018) and applies it for fact verification. *LM as Fact Checker* (Lee et al., 2020b) (U4) leverages the implicit knowledge stored in the pretrained BERT language model to verify a claim. The implementation details are given in Appendix C.

3.1 Main Results

Table 1 summarizes the fact verification performance, measured by the macro Precision (P), Recall (R), and F1 Score (F_1).

	Model	FEVER -Symmetric	FEVER-S/R	FEVER-S/R/N
		P / R / F_1	P / R / F_1	P / R / F_1
Supervised	S1. BERT-base (Devlin et al., 2019)	81.5 / 81.3 / 81.2	92.8 / 92.6 / 92.6	85.7 / 85.6 / 85.6
	S2. RoBERTa-large (Liu et al., 2019)	**85.5 / 85.5 / 85.5**	**95.2 / 95.1 / 95.1**	**88.0 / 87.9 / 87.8**
Zero-shot	U1. Random Guess	50.0 / 50.0 / 50.0	50.0 / 50.0 / 50.0	33.3 / 33.3 / 33.3
	U2. GPT2 Perplexity	52.7 / 52.7 / 52.7	55.6 / 55.6 / 55.6	35.3 / 35.3 / 35.3
	U3. MNLI-Transfer	62.2 / 55.5 / 58.7	63.6 / 60.5 / 61.8	41.4 / 39.6 / 40.7
	U4. LM as Fact Checker (Lee et al., 2020b)	71.2 / 64.5 / 67.8	77.9 / 65.6 / 70.2	64.3 / 54.6 / 49.8
	U5. QACG (BERT-base)	73.2 / 73.0 / 72.9	74.2 / 74.0 / 74.1	56.5 / 55.7 / 55.9
	U6. QACG (RoBERTa-large)	**77.3 / 77.0 / 77.1**	**78.1 / 78.1 / 78.1**	**64.6 / 62.0 / 62.6**

Table 1: Fact verification performance for supervised models and zero-shot models on three different settings.

Comparison with supervised settings. The zero-shot setting with RoBERTa-large (U6) attains 78.1 F_1 on the FEVER-S/R and 62.6 F_1 on the FEVER-S/R/N. The F_1 gap to the fully-supervised RoBERTa-large (S2) is only 17.0 and 15.2 on these two settings, respectively. These results demonstrate the effectiveness of QACG in generating good (evidence, claim) pairs for training the fact verification model. The RoBERTa model (S2, U6) is more effective than the BERT model (S1, U5) for both the zero-shot and the supervised setting.

Comparison with zero-shot baselines. Our model (U6) achieves the best results among all the zero-shot baselines across all three test sets. We find that validating a claim by its perplexity (U2) only works slightly better than random guess (U1) (+3.43 F_1), showing that misinformation does not necessary to have high perplexity. Although natural language inference seems highly correlated with fact verification, directly transferring the model trained on the MNLI dataset (U3) only outperforms random guess by 9.30 F_1. We believe this is due to the domain gap between FEVER (from Wikipedia) and the MNLI (from fiction, letters, etc.) dataset. As a generation framework, our model can avoid the domain gap issue by generating pseudo training data from the same domain (Wikipedia). Another reason is the "task gap" between NLI and fact verification, in which the former makes inference about the situation described in a sentence, while the latter focuses on claims about entities in Wikipedia.

Model Robustness. We observe a large performance drop when the supervised model is evaluated on the FEVER-Symmetric test set for both the BERT model (-11.4 F_1) and the RoBERTa model (-9.6 F_1). However, the models trained with our generated data (U2, U3) drop only 1.2 and 1.0 F_1 drop. This suggests that the wide range of different claims we generate as training data helps eliminate some of the annotation artifacts present in FEVER, leading to a more robust fact verification model.

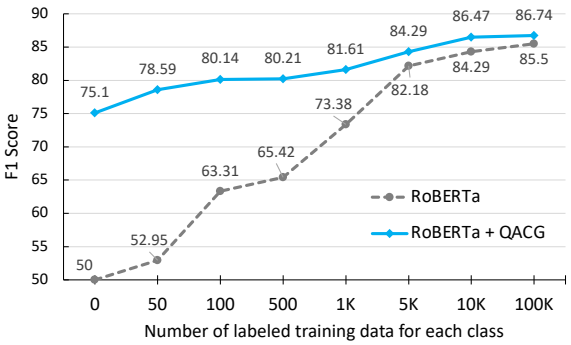


Figure 2: The few-shot learning experiment. The figure shows the F_1 score on FEVER-Symmetric for progressively larger training dataset sizes.

3.2 Few-shot Fact Verification

We then explore QACG's effectiveness in the few-shot learning setting where only a few human-labeled (evidence, claim) pairs are available. We first train the RoBERT-large fact verification model with our generated dataset QACG-Filtered. Then we fine-tune the model with a limited amount of human-labeled claims in FEVER. The blue solid line in Figure 2 shows the F_1 scores on FEVER-Symmetric after finetuning with different numbers of labeled training data. We compare this with training the model from scratch with the human-labeled data (grey dashed line).

Our model performs consistently better than the model without pretraining, regardless of the amount of labeled training data. The improvement is especially prominent in data-poor regimes; for example, our approach achieves 78.6 F_1 with only 50 labeled claims for each class, compared with 52.9 F_1 without pretraining (+25.7). This only leaves a 7.9 F_1 gap to the fully-supervised setting (86.5 F_1) with over 100K training samples. The results show pretraining fact verification with QACG

Evidence	Generated Claim
Budapest is cited as one of the most beautiful cities in Europe, ranked as the most liveable Central and Eastern European city on EIU's quality of life index, ranked as "the world's second best city" by Conde Nast Traveler, and "Europe's 7th most idyllic place to live" by Forbes.	**SUPPORTED claims** Budapest is ranked as the most liveable city in central Europe. Budapest ranks 7th in terms of idyllic places to live in Europe. **REFUTED claims** Budapest ranks in 11th in terms of idyllic places to live in Europe. Budapest is ranked the most liveable city in Asia. **NEI claims** Budapest is one of the largest cities in the European Union. Budapest is the capital of Hungary.
Alia Bhatt received critical acclaim for portraying emotionally intense characters in the road drama Highway (2014), which won her the Filmfare Critics Award for Best Actress, and the crime drama Udta Punjab (2016), which won her the Filmfare Award for Best Actress.	**SUPPORTED claims** Bhatt won the Filmfare Award for Best Actress in Udta Punjab. Bhatt received the Filmfare Critics Award for her role in Highway. **REFUTED claims** Alia Bhatt won the Best Original Screenplay award in Highway. 2 States (2014) won Alia Bhatt the Filmfare Award for Best Actress. **NEI claims** Alia Bhatt made her acting debut in the 1999 thriller Sangharsh. Bhatt played her first leading role in Karan Johar's romantic drama.

Table 2: Examples of evidence and claims generated by QACG, categorized by class labels. In the evidence, the identified answers for question generation are highlighted in blue. For claims, the correct answers are highlighted in blue for SUPPORTED claims and the replaced wrong answers are in red for REFUTED claims.

Evidence:	Roman Atwood is best known for his vlogs, where he posts updates about his life.
Claim:	Roman Atwood is a content creator.
Evidence:	In 2004, Slovenia entered NATO and the European Union.
Claim:	Slovenia uses the euro.
Evidence:	He has traveled to Chad and Uganda to raise awareness about conflicts in the regions.
Claim:	Ryan Gosling has been to a country in Africa.

Table 3: Examples of claims in FEVER that require commonsense or world knowledge (underlined).

greatly reduces the demand for in-domain human-annotated data. Our method can provide a "warm start" for fact verification system when applied to a new domain where training data are limited.

3.3 Analysis of Generated Claims

Table 2 shows representative claims generated by our model. The claims are fluent, label-cohesive, and exhibit encouraging language variety. However, one limitation is that our generated claims are mostly *lack of deep reasoning over the evidence*. This is because we finetune the question generator on the SQuAD dataset, in which more than 80% of its questions are shallow factoid questions.

To better understand whether this limitation brings a domain gap between the generated claims and the human-written claims, we randomly sampled 100 `supported` claims and 100 `refuted` and analyze whether reasoning is involved to verify those claims. We find that 38% of the `supported` claims and 16% of the `refuted` claims in FEVER require either commonsense reasoning or world knowledge to verify. Table 3 show three typical examples. Therefore, we believe this domain gap is the main bottleneck of our system. Future studies are required to generate more complex claims which involves multi-hop, numerical, and commonsense reasoning, such that we can apply our model to more complex fact checking scenario.

4 Conclusion and Future Work

We utilize the question generation model to ask different questions for given evidence and convert question–answer pairs into claims with different labels. We show that the generated claims can train a well-performing fact verification model in both the zero-shot and the few-shot learning setting. Potential future directions could be: 1) generating more complex claims that require deep reasoning; 2) extending our framework to other fact checking domains beyond Wikipedia, *e.g.*, news, social media; 3) leveraging generated claims to improve the robustness of fact checking systems.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. The UCSB authors are not supported by any of the projects above. They thank Google, Amazon, Facebook, and JP Morgan for their generous support.

Ethical Considerations

We discuss two potential issues of claim generation, showing how our work sidesteps these issues. While individuals may express harmful or biased claims, our work only focuses on generating factoid claims from a corpus. In this work, we take Wikipedia as the source for objective fact. Practicing this technique thus requires the identification of an appropriate source of objective truth to generate claims from. Another potential misuse of claim generation is to generate `refuted` claims and subsequently spread such misinformation. We caution practitioners to treat the generated claims with care. In our case, we use the generated claims only to optimize for the downstream fact verification task. We advise against releasing generated claims for public use — especially on public websites, where they may be crawled and then subsequently used for inference. As such, we will release the model code but not the output in our work. Practitioners can re-run the training pipeline to replicate experiments accordingly.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6168–6173.

Pepa Atanasova, Dustin Wright, and Isabelle Augenstein. 2020. Generating label cohesive and well-formed adversarial claims. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3168–3177.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 13042–13054.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 687–697.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT@ACL)*, pages 228–231.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2020a. Misinformation has high perplexity. *CoRR*, abs/2006.04666.

Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020b. Language models as fact checkers? *CoRR*, abs/2006.04102.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020a. Asking questions the human way: Scalable question-answer generation from text corpus. In *International World Wide Web Conference (WWW)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020b. Fine-grained fact verification with kernel graph attention network. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7342–7351.

Christopher Malon. 2019. Team papelo: Transformer networks at FEVER. *CoRR*, abs/1901.02534.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1463–1475.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Tal Schuster, Darsh J. Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3417–3423.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 809–819.

Andrew Trask, Phil Michalak, and John Liu. 2015. sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings. *CoRR*, abs/1511.06388.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5008–5020.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 1112–1122.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3901–3910.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6170–6180.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: graph-based evidence aggregating and reasoning for fact verification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 892–901.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *CCF International Conference of Natural Language Processing and Chinese Computing (NLPCC)*, pages 662–671.

A Evaluation of Question Generation

To implement the question generator, we finetune the pretrained BART model provided by HuggingFace library on the SQuAD dataset. The codes are based on the SimpleTransformers[2] library. The success of our QACG framework heavily rely on whether we can generate fluent and answerable questions given the evidence. Therefore, we separately evaluate the question generator using both automatic and human evaluation and investigate its impact to zero-shot fact verification.

A.1 Automatic Evaluation

We employ BLEU-4 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin, 2004) to evaluate the performance of our implementation. We compare the BART model with several state-of-the-art QG models, using their reported performance on the Zhou split of SQuAD.

Table 4 shows the evaluation results comparing against all baseline methods. The BART model achieves a BLEU-4 of 21.32, outperforming NQG++, S2ga-mp-gsa, and CGC-QG by large margins. This is as expected since these three baselines are based on Seq2Seq and do not apply language model pretraining. Compared with the current state-of-the-art model UniLM, the BART model achieves comparable results, with slightly lower BLEU-4 but higher METEOR.

Model	B4	MR	R_L
NQG++ (Zhou et al., 2017)	13.5	18.2	41.6
S2ga-mp-gsa (Zhao et al., 2018)	15.8	19.7	44.2
CGC-QG (Liu et al., 2020a)	17.6	21.2	44.5
UniLM (Dong et al., 2019)	**23.8**	25.6	**52.0**
BART (Lewis et al., 2020)	21.3	**27.1**	43.6

Table 4: Performance evaluation of the *Question Generator* with different model implementations. We adopt the BART model in our QACG framework. *B4*: BLEU-4, *MR*: METEOR, R_L: ROUGE-L.

A.2 Impact of Answerability

Given the evidence P and the answer A, the generated question Q must be answerable by P and

[2]https://github.com/ThilinaRajapakse/simpletransformers

Model	Answerable Rate	FV Performance P / R / F_1
NQG++	63.0%	62.2 / 62.4 / 62.3
BART	89.5%	76.3 / 76.0 / 76.1

Table 5: *Answerable Rate*: the ratio of answerable questions generated by the NQG++ and the BART model. *FV Performance*: the zero-shot fact verification performance on the FEVER-Symmetric.

Dataset		Supported	Refuted	NEI
FEVER	Train	80,035	29,775	35,517
	Test	6,666	6,666	6,666
FEVER-Symmetric		710	710	–
QACG	Full	176,370	360,924	258,452
	Filtered	100,000	100,000	100,000

Table 6: Basic statistics of the FEVER dataset and the dataset generated by QACG.

take A as its correct answer. This is the premise of generating a correct SUPPORTED claim. Therefore, we specially evaluate this *answerability* property via human ratings. We randomly sample 100 generated question-answer pairs with their corresponding evidence and ask two workers to judge the answerability of each sample. We do this for both the NQG++ model and the BART model. To investigate the impact of question quality on the fact verification performance, we separately use the NQG++ and BART as the question generator to generate claims and train the RoBERTa model. The performance is summarized in Table 5.

We find that the ratio of answerable questions generated by the BART model is 89.5%, significantly outperforms the 63.5% achieved by the NQG++ model. When switching the question generator to NQG++, the fact verification F_1 drops to 62.3 (-22.1% compared with BART). This shows that answerability plays an important role in ensuring the validity of the generated claims and has a huge impact on the fact verification performance.

B Dataset Statistics

Table 6 shows the basic data statistics of the FEVER, FEVER-Symmetric, and our generated dataset by QACG. We use the balanced dataset QACG-*Filtered* sampled from QACG-*Full* to train the fact verification model in the zero/few-shot setting. Compared with the original FEVER dataset, our generated QACG-*Filtered* dataset has a balanced number of claims for each class. Moreover, because QACG can generate three different types of claims for the same given evidence (shown in Figure 1), it results in a more "unbiased" dataset in which the model must rely on the *(evidence, claim)* pair rather than the *evidence* itself to make an inference of the class label.

C Model Implementation Details

BERT-base and RoBERTa-large (S1, S2, U5, U6). We use the `bert-base-uncased` (110M parameters) and the `roberta-large` (355M parameters) model provided by HuggingFace library to implement the BERT model and the RoBERTa model, respectively. The model is fine-tuned with a batch size of 16, learning rate of 1e-5 and for a total of 5 epochs, where the epoch with the best performance is saved.

GPT2 Perplexity (U2). To measure the perplexity, we use the HuggingFace implementation of the medium GPT-2 model (`gpt2-medium`, 345M parameters). We then rank the claims in the FEVER test set by their perplexity under the GPT-2 model. We then predict the label for each claim based on the assumption that misinformation has high perplexity. However, manually setting the perplexity threshold is difficult. Since the FEVER test set contains an equal number of claims for each class, we predict the claims in the top 1/3 of the ranking list as `refuted`, and the bottom 1/3 as `supported`. The rest claims are set as `NEI`. Therefore, the number of predicted labels for each class is also equal.

MNLI-Transfer (U3). We use the HuggingFace – BERT base model (110M parameters) fine tuned on the Multi-Genre Natural Language Inference (MNLI) corpus[3], a crowd-sourced collection of 433K sentence pairs annotated with textual entailment information. We then directly apply this model for fact verification in the FEVER test set. The class label `entailment`, `contradiction`, and `neutral` in the NLI task is mapped to `supported`, `refuted`, and `NEI`, respectively, for the fact verification task.

LM as Fact Checker (U4). Since there is no public available code for this model, we implement our own version following the settings described in Lee et al. (2020b). We use HuggingFace's `bert-base` as the language model to predict the masked named entity, and use the NLI model described in U3 as the entailment model.

[3]https://huggingface.co/textattack/bert-base-uncased-MNLI

Thank you BART!
Rewarding Pre-Trained Models Improves Formality Style Transfer

Huiyuan Lai, Antonio Toral, Malvina Nissim
CLCG, University of Groningen / The Netherlands
{h.lai, a.toral.ruiz, m.nissim}@rug.nl

Abstract

Scarcity of parallel data causes formality style transfer models to have scarce success in preserving content. We show that fine-tuning pre-trained language (GPT-2) and sequence-to-sequence (BART) models boosts content preservation, and that this is possible even with limited amounts of parallel data. Augmenting these models with rewards that target style and content –the two core aspects of the task– we achieve a new state-of-the-art.

1 Introduction and Background

Style transfer is the task of automatically converting a text of one style into another, such as turning the formal "*I viewed it and I believe it is a quality program.*" into the informal "*I've watched it and it is AWESOME!!!!*". This task, which can be used for, e.g., personalised response generation, translation of ancient text into modern text, and text simplification, is particularly challenging since style must be changed while ensuring that content is preserved. Accordingly, the performance of style transfer systems is commonly assessed on both style strength and content preservation.

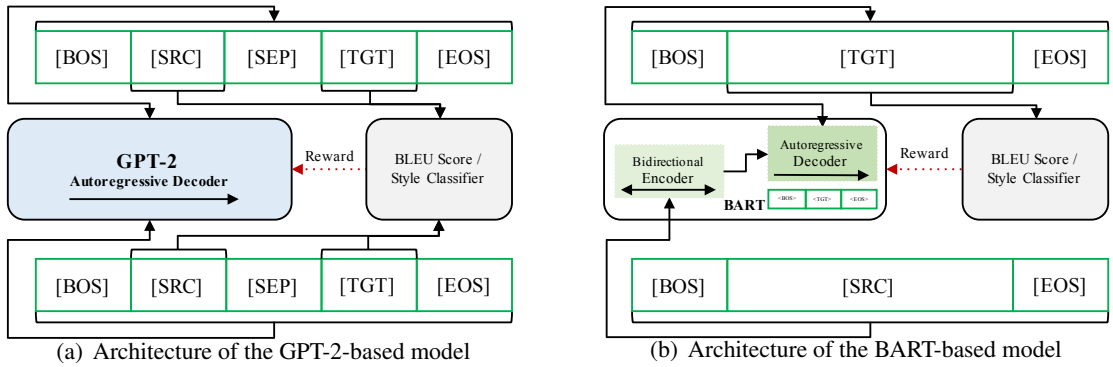
Due to the general scarcity of parallel data, unsupervised approaches are popular. These include disentangling style and content by learning a distinct representation for each (Shen et al., 2017; Fu et al., 2018; John et al., 2019), and back translation (Zhang et al., 2018; Lample et al., 2019; Luo et al., 2019; Prabhumoye et al., 2018). A common strategy to enhance style accuracy is to introduce a reward in the form of a style classifier (Lample et al., 2019; Gong et al., 2019; Luo et al., 2019; Wu et al., 2019; Sancheti et al., 2020). As a result, unsupervised models achieve good accuracy in style strength. Content preservation is however usually unsuccessful (Rao and Tetreault, 2018).

Parallel data can help to preserve content, but is limited. Niu et al. (2018) combine the train sets of two different domains and incorporate machine translation to train their models with a multi-task learning schema, plus model ensembles. Sancheti et al. (2020) use it to train a supervised sequence-to-sequence model, and in addition to the commonly used style strength reward, they include a reward based on BLEU (Papineni et al., 2002) to enhance content preservation. Shang et al. (2019) propose a semi-supervised model combining parallel data with large amounts of non-parallel data.

Pre-trained models, successful in a variety of NLP tasks, have recently been used in formality style transfer. Zhang et al. (2020) propose several data augmentation methods for pre-training a transformer-based (Vaswani et al., 2017) model and then used gold data for fine-tuning. Using GPT-2 (Radford et al., 2019), Wang et al. (2019) and Wang et al. (2020) propose a harness-rule-based preprocessing method, and joint training of bi-directional transfer and auto-encoding with two auxiliary losses. Contemporary work by Chawla and Yang (2020) develops a semi-supervised model based on BART large (Lewis et al., 2020).

Contributions Focusing specifically on *formality transfer*, for which parallel data is available, (i) we take the contribution of pre-trained models a step further by augmenting them with reward strategies that target content and style, thereby achieving new state-of-the-art results. (ii) We analyse separately the contribution of pre-trained models on content and style, showing that they take care of preserving content (the hardest part of style transfer to date), while ensuring style strength. (iii) Moreover, experimenting with training size, we show that while parallel data contributes to content preservation, fine-tuning pre-trained models with 10% of parallel data is more successful than training on 100% of data from scratch. Reducing the need for parallel data opens up the applicability of

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 484–494
August 1–6, 2021. ©2021 Association for Computational Linguistics



(a) Architecture of the GPT-2-based model

(b) Architecture of the BART-based model

Figure 1: Model architectures. We use three special symbols: [BOS] in front of every source sentence, [SEP] between the source and target sentences (only in GPT-2), and [EOS] at the end of every target sentence.

supervised style transfer to new scenarios: tasks, domains, languages.[1]

2 Method

We propose a framework to control the style of output text for style transfer atop pre-trained models. Given a source sentence $\boldsymbol{x} = \{x_1, \cdots, x_n\}$ of length n with style s_1 and a target style sentence $\boldsymbol{y} = \{y_1, \cdots, y_m\}$ of length m with style s_2, our model aims to learn two conditional distributions, altering the style of a sentence while preserving its original content. Our framework consists of (i) fine-tuning pre-trained models on a formality transfer parallel corpus; (ii) incorporating rewards to enhance style change and content preservation.

2.1 Models

GPT-2 This model (Radford et al., 2019) is a transformer-based network (Vaswani et al., 2017). Given a sentence of tokens $\boldsymbol{x} = \{x_1, \cdots, x_l\}$, the standard language modeling objective is to minimize the following negative log likelihood:

$$L_(\phi) = -\Sigma_i \log(p(x_i|x_{i-k:i-1}; \phi)) \quad (1)$$

where k is the size of the context window.

To make GPT-2 rephrase a text in the target style, the input pair ⟨Source Sentence, Target Sentence⟩ is represented as a single sequence with three special tokens to mark beginning [BOS] and end [EOS] of every sequence, and to separate source and target sentences [SEP] (Fig. 1(a)). During inference, we feed to GPT-2 the source sentence with [BOS] and [SEP] to infer the target sentence.

[1]All code at https://github.com/laihuiyuan/Pre-trained-formality-transfer.

BART This is a denoising autoencoder for pre-training sequence-to-sequence models (Lewis et al., 2020). Given a source sentence $\boldsymbol{x}$ and a target sentence $\boldsymbol{y}$, the loss function is the cross-entropy between the decoder's output and the target sentence:

$$L_(\phi) = -\Sigma_i \log(p(y_i|y_{1:i-1}, \boldsymbol{x}; \phi)) \quad (2)$$

2.2 Rewards

Atop the models, we implement two rewards, used in isolation and together, to enhance style strength (Style Classification Reward) and content preservation (BLEU Score Reward).

Style Classification Reward As often done in previous work (see Section 1), we use a classification confidence reward to encourage larger change in the confidence of a style classifier (SC). We pre-train the binary style classifier TextCNN (Kim, 2014) and use it to evaluate how well the transferred sentence $\boldsymbol{y}'$ matches the target style. SC's confidence is formulated as

$$p(s_i|\boldsymbol{y}') = softmax_i(TextCNN(\boldsymbol{y}', \theta)) \quad (3)$$

where i = {1,2}, and represent source and target style respectively. θ are the parameters of the style classifier, fixed during fine-tuning. The reward is

$$R_{cls} = \lambda_{cls}[p(s_2|\boldsymbol{y}') - p(s_1|\boldsymbol{y}')] \quad (4)$$

where $\boldsymbol{y}'$ is the generated target sentence sampled from the model's distribution at each time step in decoding. For the GPT-2 based model, we also add a classification confidence reward to the source sentence, similar to Eq. 4, since the model generates sentence $\boldsymbol{x}'$ with the original style while generating the target sentence:

$$R_{cls_{source}} = \lambda_{cls}[p(s_1|\boldsymbol{x}') - p(s_2|\boldsymbol{x}')] \quad (5)$$

		0 ⟶ 1		1 ⟶ 0	
Domain	Train	Valid	Test	Valid	Test
F&R	51,967	2,788	1,332	2,247	1,019
E&M	52,595	2,877	1,416	2,356	1,082

Table 1: GYAFC dataset. 0 = informal; 1 = formal.

BLEU Score Reward Following Sancheti et al. (2020), we introduce a BLEU-based reward to foster content preservation as in Eq. 6, where $\boldsymbol{y}'$ is the target style text obtained by greedily maximizing the distribution of model outputs at each time step, and $\boldsymbol{y}^s$ is sampled from the distribution.

$$R_{bleu} = \lambda_{bleu}[bleu(\boldsymbol{y}', \boldsymbol{y}) - bleu(\boldsymbol{y}^s, \boldsymbol{y})] \quad (6)$$

Gradients and Objectives The rewards are used for policy learning. The policy gradient[2] is

$$\nabla_\phi J(\phi) = E[R \cdot \nabla_\phi log(P(\boldsymbol{y}^s|\boldsymbol{x}; \phi))] \quad (7)$$

where R is the SC reward and/or the BLEU reward, $\boldsymbol{y}^s$ is sampled from the distribution of model outputs at each decoding time step, and ϕ are the parameters of the model. Similarly, we add the policy gradient regarding the source sentence for the SC reward (only for the GPT-2-based model).

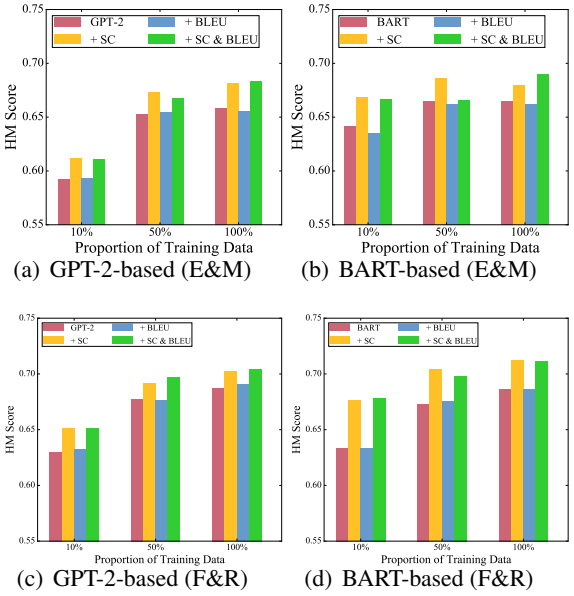
The overall objectives for ϕ are the loss of the base model (Eq. 1 or Eq. 2) and the policy gradient of the different rewards (Eq. 7).

3 Experiments

Dataset Grammarly's Yahoo Answers Formality Corpus (GYAFC) (Rao and Tetreault, 2018) is a formality style transfer dataset with parallel formal and informal sentences from two domains: Entertainment & Music (E&M) and Family & Relationships (F&R). Table 1 shows the number of sentences in train, validation, and test. Four human references exist for every valid/test sentence.

Setup All experiments are implemented atop Huggingface's transformers (Wolf et al., 2020). Our base models are the GPT-2-based model (117M parameters) and BART-based model (base with 139M parameters and large with 406M). We fine-tune them with the Adam optimiser (Kingma and Ba, 2015) with batch size 32; the initial learning rates are $5e^{-5}$ (GPT-2) and $3e^{-5}$ (BART). The final values for λ are set to 1 for SC and 0.2 for BLEU based on validation results. We use early stopping (patience 3) if validation performance does not improve. Test results are reported with the best validation settings.

[2]Additional details are provided in the Appendix.



(a) GPT-2-based (E&M) (b) BART-based (E&M)

(c) GPT-2-based (F&R) (d) BART-based (F&R)

Figure 2: HM score of x%-sized training sets of GPT-2-/BART-based models with different rewards (none, +SC, +BLEU, +SC & BLEU) for the two domains (E&M and F&R).

Evaluation Following previous work (Luo et al., 2019; He et al., 2020; Sancheti et al., 2020), we adopt the following strategies. The binary classifier TextCNN (Kim, 2014) is pre-trained to evaluate style strength; on the human references it has an accuracy of 87.0% (E&M) and 89.3% (F&R). Based on the four human references, we calculate BLEU[3] for content preservation. As overall score we compute the harmonic mean (HM) of style accuracy and BLEU. For our evaluation we also test BLEURT, a recent metric for content preservation which correlates better with human judgments than other metrics that take semantic information into account, e.g. METEOR (Sellam et al., 2020).

Baselines We train a basic supervised model (a Bi-LSTM with attention from OpenNMT (Klein et al., 2017)), to assess the impact of the size of parallel training data. We compare our models to the five baselines from Rao and Tetreault (2018), and to the best performing formality style transfer methods that report results on the datasets we use. These are mentioned in Section 1 and summarised as follows: Bi-directional FT (Niu et al.,

[3]We use `multi-bleu.perl` with default settings.

Domain	Model	BLEURT	BLEU	ACC	HM	Model	BLEURT	BLEU	ACC	HM
E&M	OpenNMT + SC & BLEU (10% data)	-0.919	0.231	0.886	0.366	OpenNMT + SC & BLEU (100% data)	-0.420	0.403	0.804	0.537
	(A) INFORMAL ↔ FORMAL					(B) INFORMAL ⟶ FORMAL				
	NMT-Combined (Rao and Tetreault, 2018)	-0.100	0.501	0.797	0.615	GPT-CAT (train on E&M and F&R, Wang et al. (2019))	0.176	0.725	0.876	0.793
	GPT-2 + SC & BLEU (10% data, **Ours**)	-0.058	0.495	0.799	0.611	Chawla's (Chawla and Yang (2020))	0.260	0.762	0.910	0.829
	GPT-2 + SC & BLEU (100% data, **Ours**)	-0.007	0.542	**0.923**	0.683	BART + SC & BLEU (train on E&M, **Ours**)	0.218	0.730	0.887	0.801
	BART + SC & BLEU(10% data, **Ours**)	-0.030	0.547	0.855	0.667	BART + SC & BLEU (train on E&M and F&R, **Ours**)	0.236	0.745	**0.937**	0.830
	BART + SC & BLEU (100% data, **Ours**)	**0.044**	**0.577**	0.859	**0.690**	BART large + SC & BLEU (train on E&M and F&R, **Ours**)	**0.274**	**0.765**	0.929	**0.839**
	(C) INFORMAL ↔ FORMAL & COMBINED DOMAINS					(D) BLEU EVALUATED AGAINST THE FIRST REFERENCE				
	Bi-directional FT (Niu et al., 2018)	0.023	0.554	0.818	0.661	*TS→CP (Sancheti et al. (2020))	-	0.292	-	-
	BART large + SC & BLEU (100% data, **Ours**)	**0.078**	**0.596**	**0.905**	**0.719**	BART + SC & BLEU (100% data, **Ours**)	-	**0.306**	-	-
F&R	OpenNMT + SC & BLEU (10% data)	-0.706	0.303	0.859	0.448	OpenNMT + SC & BLEU (100% data)	-0.304	0.477	0.789	0.595
	(A) INFORMAL ↔ FORMAL					(B) INFORMAL ⟶ FORMAL				
	NMT-Combined (Rao and Tetreault, 2018)	-0.089	0.527	0.798	0.635	*GPT-CAT (train on E&M and F&R, Wang et al. (2019))	-	0.769	-	-
	GPT-2 + SC & BLEU (10% data, **Ours**)	-0.027	0.528	0.849	0.651	Chawla's (Chawla and Yang (2020))	0.302	**0.799**	0.910	0.851
	GPT-2 + SC & BLEU (100% data, **Ours**)	0.038	0.572	**0.915**	0.704	BART + SC & BLEU (train on F&R, **Ours**)	0.271	0.770	0.897	0.829
	BART + SC & BLEU (10% data, **Ours**)	0.039	0.571	0.833	0.678	BART + SC & BLEU (train on F&R and E&M, **Ours**)	0.270	0.777	0.912	0.839
	BART + SC & BLEU (100% data, **Ours**)	**0.068**	**0.595**	0.882	**0.711**	BART large + SC & BLEU (train on F&R and E&M, **Ours**)	**0.324**	0.793	**0.920**	**0.852**
	(C) INFORMAL ↔ FORMAL & COMBINED DOMAINS					(D) 10% PARALLEL TRAINING DATA				
	Bi-directional FT (Niu et al. (2018)	0.037	0.568	0.839	0.677	*CPLS (Shang et al., 2019)	-	0.379	-	-
	BART large + SC & BLEU (100% data, **Ours**)	**0.100**	**0.611**	**0.900**	**0.728**	BART + SC & BLEU (**Ours**)	-	**0.571**	-	

Table 2: Comparison of our models to previous work. The best score for each metric in each block is boldfaced. Notes: (i) if the output of previous work is available, we re-calculate the scores using our evaluation metrics. Otherwise, scores are from the paper and we mark this with (*); (ii) (B) shows our results on informal-to-formal to compare with Wang et al. (2019) and Chawla and Yang (2020), who only transfer in this direction; (iii) in (C) we train on the concatenated data from both domains, to compare against Niu et al. (2018); (iv) in (E&M (D)) we re-evaluate our system against the first reference only, as done by Sancheti et al. (2020).

2018), CPLS (Shang et al., 2019), GPT-CAT (Wang et al., 2019), S2S-SLS (GPT-2) (Wang et al., 2020), Transformer (data augmentation) (Zhang et al., 2020), TS→CP (Sancheti et al., 2020), and Chawla's (Chawla and Yang, 2020). Since supervised methods significantly outperform unsupervised approaches, results for the latter are not considered as the baseline in our experiment.. Disentanglement-based methods are not included since Lample et al. (2019) provide evidence that they are surpassed.

Results Figure 2 shows the HM score of x%-sized training sets on the E&M and the F&R domains. Increasing train set size from 10% to 50% has a greater boost on GPT-2-based models than BART's. However, BART-based models obtain the highest results. Table 2 reports a selection of our models [4] and previous state-of-the-art work. Zooming in on the single measures, we see in Table 2 how varying training size reveals the impact of parallel data on content preservation: OpenNMT's BLEU score on E&M increases from 0.231 with 10% of the data to 0.403 with 100%. Style accuracy appears instead easier to achieve even with limited supervision. Increasing training size for fine-tuning either pre-trained model does not however yield dramatic improvements in content preservation (e.g. from 0.547 to 0.577 BLEU for BART base on E&M). In fact, fine-tuning a pre-trained model (either GPT-2 or BART) with just 10% of parallel data, leads to better content preservation (0.547 BLEU with BART on E&M) than OpenNMT with 100% (0.403). This suggests that content preservation is largely taken care of by the pre-trained models, already, and can explain why the BLEU-based reward does not help too much in isolation (see Fig. 2). Conversely, the SC reward consistently boosts style accuracy in both BART and GPT-2. Nevertheless, combining rewards can be beneficial. Overall, BART-based models perform better on content preservation while results on style strength are mixed.

Given the experimental setup of some previous work, we ran additional comparisons (blocks (B), (C), and (D) of Table 2). In all cases, our results are higher than the previous state-of-the-art. For example, in F&R (D) our model with 10% parallel data outperforms Shang et al. (2019)'s semi-supervised model, which uses about 9.5% parallel data and large amounts of non-parallel data (BLEU 0.571 vs 0.379). Fine-tuning BART on both domains (C)[5] leads to the best results to date on both datasets (E&M: 0.719; F&R: 0.728).

With respect to the two evaluation metrics used for content preservation (BLEU and BLEURT), we can observe in Table 2 that they follow a similar trend. In fact, they correlate very highly (Pearson's $r = .951$, p<.001, $n = 14$ for E&M, and $r = .951$,

[4] In the table we report results for the models that use both rewards (BLEU and SC) since this setting mostly leads to best results. Complete results for all models (and sample outputs) are in the Appendix.

[5] Following Kobus et al. (2017), we add a token to each training instance that specifies its domain.

System	Sentence	BLEURT	BLEU	ACC
FROM INFORMAL TO FORMAL				
Source	i say omarion.he has the hair clothes and body,a triple deal on one person.		-	
Reference 1	My choice is Omarion as he has high quality, hair, clothes, and body to create a triple deal in one person.		-	
Reference 2	I would say Omarion because he has the hair, clothes, and body; A triple deal on a single person.		-	
Reference 3	I pick Omarion, he has the hair, the clothes, and the body. A triple deal on one person.		-	
Reference 4	Omarion has the hair, clothes, and the body.		-	
PBMT-Combined (Rao and Tetreault, 2018)	I say omarion. he has the hair, clothes and body, the deal on one person.	-0.153	0.509	0.946
Bi-directional FT (Niu et al., 2018)	I say Omarion, he has the hair clothes and body, and a triple deal on one person.	-0.149	0.510	0.953
GPT-CAT (Wang et al., 2019)	I say Omarion. He has the hair, clothes, and body, a triple deal on one person.	0.044	0.585	**1.000**
S2S-SLS (Wang et al., 2020)	I say Omarion. He has the hair clothes and body, a triple deal on one person.	-0.035	0.350	**1.000**
Transformer (Zhang et al., 2020)	I say omarionhe has the hair clothes and body, a triple deal on one person.	-0.255	0.462	0.892
Chawla's (Chawla and Yang, 2020)	I say Marion because he has the hair, clothes and body, a triple deal on one person.	-0.538	0.534	0.989
OpenNMT + SC & BLEU (**Ours**)	I say Omarion. He has the hair clothes and body.	-0.325	0.147	**1.000**
GPT-2 + SC & BLEU (**Ours**)	I say Omarion. He has the hair clothes and body, a triple deal on one person.	-0.035	0.350	**1.000**
BART base + SC & BLEU (**Ours**)	I would say Omar. He has the hair, clothes, and body. It is a triple deal on one person.	-0.012	0.589	**1.000**
BART large + SC & BLEU (**Ours**)	I would say Omarion. He has the hair, clothes, and body, a triple deal on one person.	**0.096**	**0.657**	**1.000**
FROM FORMAL TO INFORMAL				
Source	I suggest avoiding hot dogs, and not watching this movie with your little sister.		-	
Reference 1	Don't eat hot dogs, or watch this movie with your little sister!		-	
Reference 2	Don't do hot dogs or this movie with your kid sister.		-	
Reference 3	don't eat hot dogs and don't watch it w/ ur lil sis!		-	
Reference 4	Don't eat hot dogs or watch this flick with your lil sis!		-	
PBMT-Combined (Rao and Tetreault, 2018)	I suggest avoiding hot dogs, and not watching this movie with your little sister.	-0.298	0.417	0.004
Bi-directional FT (Niu et al., 2018)	I suggest avoiding hot dogs and not watching this movie with your little sister.	-0.233	0.437	0.009
OpenNMT with SC & BLEU	Can't watch this movie with your little sister.	-0.521	0.542	0.783
GPT-2 + SC & BLEU	don't watch this movie with your little sister.	-0.415	0.599	**1.000**
BART + SC & BLEU	avoid hot dogs and not watch this movie with your little sister.	**-0.016**	0.610	0.925
BART large + SC & BLEU	Avoid hot dogs and don't watch this movie with your little sister.	-0.171	**0.800**	0.825

Table 3: Sample model outputs and their sentence-level scores on the E&M domain, where red denotes improperly generated words or content. Note that ACC indicates style confidence here.

$p<.001$, $n = 13$ for F&R).

Finer-grained Analysis Table 3 shows example outputs and their evaluation according to the metrics we use; the outputs are produced by existing systems we compare to, and our own models.[6]

In the "Informal to Formal" example, we can see that text generated by most systems is assessed with a high confidence in style conversion, except for PBMT-Combined (Rao and Tetreault, 2018) and Transformer (Zhang et al., 2020) (the name "omarionhe" should be "Omarion", and the word "he" at the beginning of the sentence should be "He"). However, the sentences generated by previous systems are not so fluent, and some of them fail in preserving content (Transformer (Zhang et al., 2020) ("omarionhe") and Chawla's (Chawla and Yang, 2020) ("Marion")). For our models, the Bi-LSTM based model fails in content preservation while the systems based on pre-trained models are much better at this task. Our model based on BART Large generates this specific sentence accurately in terms of content preservation, style strength, and fluency.

When looking at the "Formal to Informal" example in Table 3, we observe that the two previously existing systems replace very little (one comma by the Bi-directional FT (Niu et al., 2018)) or nothing at all (PBMT-Combined (Rao and Tetreault, 2018)). Conversely, our systems make substantial modifications, resulting in output sentences that are noticeably more informal than the input sentence. OpenNMT and the GPT-2-based models lose part of the content (the suggestion to avoid hot dogs) while the two BART-based systems manage to preserve the whole message.

[6]More examples are in Appendix.

4 Conclusions

Fine-tuning pre-trained models proves a successful strategy for formality style transfer, especially towards content preservation, thereby reducing the need for parallel data. A sequence-to-sequence pre-trained model (BART) outperforms a language model (GPT-2) in content preservation, and overall, and with the addition of rewards achieves new state-of-the-art results. The fact that GPT-2 is instead often better at style strength could be (partly) due to how the style reward is implemented in the two models (Eq. 4 and 5), and will need further investigation. For a better understanding of the different behaviour of BART and GPT-2 for this task, the next natural step is to include human evaluation.

Acknowledgments

This work was partly funded by the China Scholarship Council (CSC). The anonymous ACL reviewers provided us with useful comments which contributed to improving this paper and its presentation, so we're grateful to them. We would also like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

Impact Statement

All work that automatically generates and/or alters natural text could unfortunately be used maliciously. While we cannot fully prevent such uses once our models are made public, we do hope that writing about risks explicitly and also raising awareness of this possibility in the general public are ways to contain the effects of potential harmful uses. We are open to any discussion and suggestions to minimise such risks.

References

Kunal Chawla and Diyi Yang. 2020. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2340–2354.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 663–670.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3168–3180.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *Proceedings of Ninth International Conference on Learning Representations*.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 372–378.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *Proceedings of Seventh International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5116–5122.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 129–140.

Abhilasha Sancheti, Kundan Krishna, Balaji Vasan Srinivasan, and Anandhavelu Natarajan. 2020. Reinforced rewards framework for text style transfer. In *Advances in Information Retrieval*, pages 545–560.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. 2019. Semi-supervised text style transfer: Cross projection in latent space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4937–4946.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6833–6844.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *In Advances in Neural Information Processing Systems*, pages 6000–6010.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wen-Han Chao. 2020. Formality style transfer with shared latent space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2236–2249, Barcelona, Spain (Online).

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883.

Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel data augmentation for formality style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation. *arXiv preprint, arXiv: 1808.07894*.

A Appendices

This Appendices include: 1) detailed results for all experiments (A.1); 2) more details on policy gradient (A.2); 3) some example outputs of various models and their sentence-level scores, to give an idea of what the generated sentences look like when style transfer is applied. We specifically focus on the 100% parallel data settings for our models (A.3).

A.1 Detailed Results of Models

We report here the full set of results for all our models and previous work.

(a) Detailed Results of Our Models

Model	BLEURT	BLEU	ACC	HM	BLEURT	BLEU	ACC	HM	BLEURT	BLEU	ACC	HM
Proportion of parallel training data	10%				50%				100%			
OpenNMT (Bi-LSTM)	-0.919	0.231	0.886	0.366	-0.489	0.392	0.789	0.524	**-0.420**	0.403	0.804	0.537
OpenNMT + SC	**-0.902**	**0.238**	**0.893**	**0.376**	-0.500	0.386	**0.821**	0.526	-0.451	0.399	0.789	0.530
OpenNMT + BLEU	-0.926	0.232	0.888	0.368	**-0.485**	0.389	0.800	0.523	-0.485	**0.412**	0.767	0.536
OpenNMT + SC & BLEU	-0.903	0.234	0.890	0.371	-0.497	**0.391**	0.813	**0.528**	-0.442	0.403	**0.810**	**0.538**
GPT-2 base	-0.042	0.492	0.741	0.592	0.004	**0.541**	0.825	0.653	**0.006**	**0.549**	0.821	0.658
GPT-2 + SC	-0.048	0.492	**0.810**	**0.612**	-0.014	0.531	**0.919**	**0.673**	-0.001	0.543	0.917	0.682
GPT-2 + BLEU	**-0.041**	**0.497**	0.735	0.593	**0.006**	0.539	0.833	0.655	0.005	0.546	0.822	0.656
GPT-2 + SC & BLEU	-0.058	0.495	0.799	0.611	-0.014	0.530	0.903	0.668	-0.007	0.542	**0.923**	**0.683**
BART base	**0.035**	**0.547**	0.776	0.642	0.036	**0.572**	0.794	0.665	0.048	**0.578**	0.784	0.665
BART + SC	0.021	0.539	**0.882**	**0.669**	0.035	0.566	**0.872**	**0.686**	0.045	0.571	0.841	0.680
BART + BLEU	0.034	0.541	0.769	0.635	0.040	0.567	0.796	0.662	**0.050**	0.576	0.777	0.662
BART + SC & BLEU	0.030	**0.547**	0.855	0.667	**0.042**	0.562	0.817	0.666	0.044	0.577	**0.859**	**0.690**
BART large + SC & BLEU	0.035	0.560	0.847	0.674	0.070	0.585	0.900	0.709	0.072	0.584	0.886	0.704
COMBINED TWO DOMAINS WITHOUT DOMAIN TAG												
BART base	**0.038**	**0.559**	0.731	0.634	0.050	**0.581**	0.795	0.671	**0.054**	**0.585**	0.809	0.679
BART + SC	0.031	0.546	**0.830**	0.659	0.043	0.575	**0.865**	**0.691**	0.039	**0.585**	**0.884**	**0.704**
BART + BLEU	0.033	0.555	0.743	0.635	0.042	0.575	0.810	0.673	**0.054**	0.583	0.814	0.679
BART + SC & BLEU	0.024	0.556	0.815	**0.661**	**0.054**	0.578	0.845	0.685	0.050	0.580	0.859	0.692
BART large + sc & BLEU	0.071	0.576	0.867	0.692	0.075	0.593	0.887	0.711	0.086	0.597	0.888	0.714
COMBINED TWO DOMAINS WITH DOMAIN TAG												
BART base	**0.042**	0.552	0.754	0.637	0.054	0.579	0.748	0.653	**0.060**	0.582	0.787	0.669
BART + SC	0.035	0.555	0.831	0.666	0.039	0.571	0.833	0.678	0.046	0.579	**0.895**	**0.703**
BART + BLEU	0.039	0.554	0.745	0.635	**0.056**	0.578	0.745	0.651	0.049	**0.588**	0.825	0.685
BART + SC & BLEU	0.039	**0.556**	**0.845**	**0.671**	0.046	**0.580**	**0.834**	**0.684**	0.047	0.583	0.883	0.702
BART large + SC & BLEU	0.077	0.575	0.793	0.667	0.073	0.587	0.870	0.701	0.078	0.596	0.905	0.719

Table A.1.1: Evaluation results of x%-sized training sets (10%, 50% and 100%) on the E&M domain. The best score for each metric in each table section is boldfaced. BLEURT scores are calculated based on the BLEURT-base model with 128 tokens. Note that (i) Both BLEURT and BLEU are calculated against the four human references; (ii) ACC is the accuracy of the output labeled as the target style by the binary classifier; and (iii) HM is the harmonic mean of ACC and BLEU.

Model	BLEURT	BLEU	ACC	HM	BLEURT	BLEU	ACC	HM	BLEURT	BLEU	ACC	HM
Proportion of parallel training data	10%				50%				100%			
OpenNMT (Bi-LSTM)	-0.706	0.303	0.859	0.448	-0.304	0.449	0.792	0.573	-0.304	0.477	0.789	0.595
OpenNMT + SC	**-0.695**	**0.322**	**0.860**	**0.469**	-0.337	0.447	0.838	**0.583**	-0.289	0.466	0.824	0.595
OpenNMT + BLEU	-0.712	0.311	0.829	0.452	**-0.292**	**0.455**	0.808	0.582	**-0.246**	**0.478**	0.789	0.595
OpenNMT + SC & BLEU	-0.699	0.320	0.828	0.462	-0.332	0.444	**0.847**	**0.583**	-0.288	0.472	**0.848**	**0.606**
GPT-2 base	-0.020	**0.531**	0.775	0.630	**0.027**	**0.567**	0.841	0.677	**0.046**	0.576	0.850	0.687
GPT-2 + SC	-0.031	0.529	0.847	**0.651**	0.020	0.563	0.897	0.692	0.031	0.569	**0.916**	0.702
GPT-2 + BLEU	**-0.016**	0.529	0.786	0.632	0.026	0.566	0.838	0.676	0.041	**0.577**	0.860	0.691
GPT-2 + SC & BLEU	-0.027	0.528	**0.849**	**0.651**	0.015	0.562	**0.917**	**0.697**	0.038	0.572	0.915	**0.704**
BART base	**0.045**	0.565	0.719	0.633	0.071	0.589	0.786	0.673	**0.080**	0.600	0.801	0.686
BART + SC	0.041	0.569	**0.833**	0.676	0.061	**0.592**	**0.869**	**0.704**	0.067	0.601	0.874	**0.712**
BART + BLEU	0.041	0.566	0.719	0.633	**0.072**	0.590	0.789	0.675	0.078	**0.602**	0.798	0.686
BART + SC & BLEU	0.039	**0.571**	**0.833**	**0.678**	0.057	0.589	0.858	0.698	0.068	0.595	**0.882**	0.711
BART large + SC & BLEU	0.095	0.585	0.816	0.681	0.087	0.604	0.891	0.720	0.095	0.615	0.876	0.722
COMBINED TWO DOMAINS WITHOUT DOMAIN TAG												
BART base	**0.035**	**0.572**	0.734	0.643	0.060	0.592	0.821	0.688	**0.074**	0.604	0.807	0.691
BART + SC	0.026	0.563	**0.821**	0.668	0.056	0.592	**0.890**	**0.711**	0.054	0.602	**0.877**	**0.714**
BART + BLEU	0.033	0.568	0.732	0.640	**0.064**	0.593	0.834	0.693	0.073	**0.606**	0.831	0.701
BART + SC & BLEU	0.028	**0.57**2	0.812	**0.671**	0.054	**0.596**	0.843	0.698	0.063	0.601	0.872	0.712
BART large + SC & BLEU	0.087	0.598	0.869	0.708	0.094	0.607	0.871	0.715	0.100	0.610	0.889	0.724
COMBINED TWO DOMAINS WITH DOMAIN TAG												
BART base	0.042	0.570	0.779	0.658	**0.072**	0.592	0.768	0.669	**0.078**	0.604	0.801	0.689
BART + SC	0.035	**0.574**	0.849	**0.685**	0.058	0.586	**0.861**	0.697	0.059	0.599	0.892	0.718
BART + BLEU	**0.047**	0.572	0.761	0.653	0.071	0.591	0.772	0.669	0.077	**0.605**	0.817	0.695
BART + SC & BLEU	0.043	0.573	**0.850**	**0.685**	0.057	**0.595**	0.849	**0.700**	0.064	0.603	**0.896**	**0.721**
BART large + SC & BLEU	0.089	0.590	0.801	0.679	0.099	0.604	0.869	0.713	0.100	0.611	0.900	0.728

Table A.1.2: Evaluation results of x%-sized training sets (10%, 50% and 100%) on the F&R domain. The best score for each metric in each table section is boldfaced. BLEURT scores are calculated based on the BLEURT-base model with 128 tokens. Note that (i) Both BLEURT and BLEU are calculated against the four human references; (ii) ACC is the accuracy of the output labeled as the target style by the binary classifier; and (iii) HM is the harmonic mean of ACC and BLEU.

(b) Comparison of our models with the other models

Domain	Model	BLEURT	BLEU	ACC	HM	Model	BLEURT	BLEU	ACC	HM
	(A) INFORMAL ↔ FORMAL					(B) INFORMAL ⟶ FORMAL				
E&M	Rule-based (Rao and Tetreault, 2018)	-0.221	0.420	0.704	0.526	GPT-CAT (train on E&M, Wang et al. (2019))	0.170	0.713	0.905	0.801
	NMT-baseline (Rao and Tetreault, 2018)	-0.267	0.437	0.851	0.577	GPT-CAT (train on E&M and F&R, Wang et al. (2019))	0.176	0.725	0.876	0.793
	NMT-copy (Rao and Tetreault, 2018)	-0.269	0.441	0.808	0.571	S2S-SLS(Wang et al. (2020))	0.173	0.711	0.919	0.802
	NMT-Combined (Rao and Tetreault, 2018)	-0.100	0.501	0.797	0.615	Transformer (Zhang et al. (2020))	0.191	0.734	0.887	0.803
	PBMT-Combined (Rao and Tetreault, 2018)	-0.088	0.502	0.753	0.602	Chawla's (Chawla and Yang, 2020)	0.260	0.762	0.910	0.829
	GPT-2 + SC & BLEU (10% data, **Ours**)	-0.058	0.495	0.799	0.611	GPT-2 + SC & BLEU (train on E&M, **Ours**)	0.159	0.701	0.927	0.798
	GPT-2 + SC & BLEU (100% data, **Ours**)	-0.007	0.542	**0.923**	0.683	BART + SC & BLEU (train on E&M, **Ours**)	0.218	0.730	0.887	0.801
	BART + SC & BLEU (10% data, **Ours**)	0.030	0.547	0.855	0.667	BART + SC & BLEU (train on E&M and F&R, **Ours**)	0.236	0.745	**0.937**	0.830
	BART + SC & BLEU (100% data, **Ours**)	**0.044**	**0.577**	0.859	**0.690**	BART large + SC & BLEU (train on E&M and F&R, **Ours**)	**0.274**	**0.765**	0.929	**0.839**
	(C) INFORMAL ↔ FORMAL & COMBINED DOMAINS					(D) BLEU EVALUATED AGAINST THE FIRST REFERENCE				
	Bi-directional FT (Niu et al., 2018)	0.023	0.554	0.818	0.661	*TS→CP (Sancheti et al., 2020)	-	0.292	-	-
	BART large + SC & BLEU (10% data, **Ours**)	0.077	0.575	0.793	0.667	GPT-2 + SC & BLEU (100% data, **Ours**)	-	0.296	-	-
	BART large + SC & BLEU (100% data, **Ours**)	**0.078**	**0.596**	**0.905**	**0.719**	BART + SC & BLEU (100% data, **Ours**)	-	**0.306**	-	-
	(A) INFORMAL ↔ FORMAL					(B) INFORMAL ⟶ FORMAL				
F&R	Rule-based (Rao and Tetreault, 2018)	-0.226	0.450	0.738	0.559	*GPT-CAT (train on F&R, Wang et al. (2019))	-	0.773	-	-
	NMT-baseline (Rao and Tetreault, 2018)	-0.183	0.500	0.818	0.621	*GPT-CAT (train on E&M and F&R, Wang et al. (2019))	-	0.769	-	-
	NMT-copy (Rao and Tetreault, 2018)	-0.186	0.492	0.807	0.611	S2S-SLS(GPT-2, Wang et al. (2020))	0.244	0.766	0.857	0.809
	NMT-Combined (Rao and Tetreault, 2018)	-0.089	0.527	0.798	0.635	Transformer (Zhang et al. (2020))	0.246	0.770	0.890	0.827
	PBMT-Combined (Rao and Tetreault, 2018)	-0.062	0.517	0.788	0.624	Chawla's (Chawla and Yang, 2020)	0.302	**0.799**	0.910	0.851
	GPT-2 + SC & BLEU (10% data, **Ours**)	-0.027	0.528	0.849	0.651	GPT-2 + SC & BLEU (train on F&R, **Ours**)	0.226	0.747	0.921	0.825
	GPT-2 + SC & BLEU (100% data, **Ours**)	0.038	0.572	**0.915**	0.704	BART + SC & BLEU (train on F&R, **Ours**)	0.271	0.770	0.897	0.829
	BART + SC & BLEU (10% data, **Ours**)	0.039	0.571	0.833	0.678	BART + SC & BLEU (train on F&R and E&M, **Ours**)	0.270	0.777	0.912	0.839
	BART + SC & BLEU (100% data, **Ours**)	**0.068**	**0.595**	0.882	**0.711**	BART large + SC & BLEU (train on F&R and E&M, **Ours**)	**0.324**	0.793	**0.920**	**0.852**
	(C) INFORMAL ↔ FORMAL & COMBINED DOMAINS					(D) 10% PARALLEL TRAINING DATA (FROM PAPER)				
	Bi-directional FT (Niu et al. (2018)	0.037	0.568	0.839	0.677	*CPLS (Shang et al., 2019)	-	0.379	-	-
	BART large + SC & BLEU (10% data, **Ours**)	0.089	0.590	0.801	0.679	GPT-2 + SC & BLEU (**Ours**)	-	0.528	-	-
	BART large + SC & BLEU (100% data, **Ours**)	**0.100**	**0.611**	**0.900**	**0.728**	BART + SC & BLEU (**Ours**)	-	**0.571**	-	-

Table A.1.3: Comparison of our models with the other models. The best score for each metric in each block is boldfaced. BLEURT scores are calculated based on the BLEURT-base model with 128 tokens. Notes: (i) if the output of a previous work is available, we re-calculate the scores using our evaluation metrics. Otherwise we take the scores from the paper and mark this with a (*); (ii) in (B) we report our results on informal-to-formal alone to compare with several systems which only transfer in this direction; (iii) in (C) we train systems on the concatenated data from both domains, to compare against Niu et al. (2018); (iv) in (E&M (D)) we re-evaluate our system against the first reference only, as this is what Sancheti et al. (2020) do.

A.2 Policy Gradient

Reinforcement learning (RL) is a sub-field of machine learning that is concerned with how intelligent agents ought to take actions in an environment in order to maximize the cumulative reward. Here, we employ the policy gradient algorithm (Williams, 1992) to maximize the expected reward (style strength and/or content preservation) of the generated sequence $\boldsymbol{y}^s$, whose gradient with respect to the parameters ϕ of the neural network model is estimated by sampling as:

$$\begin{aligned}
\nabla_\phi J(\phi) &= R \cdot \nabla_\phi \sum_i P(\boldsymbol{y}_i^s|\boldsymbol{x}_i;\phi) \\
&= \sum_i P(\boldsymbol{y}_i^s|\boldsymbol{x}_i;\phi) R_i \nabla_\theta \log(P(\boldsymbol{y}_i^s|\boldsymbol{x}_i;\phi)) \\
&\simeq \frac{1}{N}\sum_{i=1}^{N} R_i \nabla_\phi \log(P(\boldsymbol{y}_i^s|\boldsymbol{x}_i;\phi)) \\
&= E[R \cdot \nabla_\phi log(P(\boldsymbol{y}^s|\boldsymbol{x};\phi))]
\end{aligned} \tag{8}$$

where $J(\cdot)$ is the objective function, $\nabla_\phi J(\cdot)$ is the gradient of $J(\cdot)$ with respect to ϕ, R_i is the reward of the i_{th} sequence $\boldsymbol{y}^s$ that is sampled from the distribution of model outputs at each decoding time step, ϕ are the parameters of the model, N is the sample size, and $E(\cdot)$ is the expectation.

Regarding the reward of style classification for GPT-2 based model, we design two rewards (Eq. 4 and Eq. 5) for source sentence and target sentence, respectively. The policy gradient is then

$$\begin{aligned}
\nabla_\phi J(\phi) = &E[R_{cls_{source}} \cdot \nabla_\phi log(P(\boldsymbol{y}^s_{source}|\boldsymbol{x}_{source};\phi))] \\
&+ E[R_{cls_{target}} \cdot \nabla_\phi log(P(\boldsymbol{y}^s_{target}|\boldsymbol{x}_{source,target};\phi))]
\end{aligned} \tag{9}$$

A.3 Example Outputs of Various Models

System	From informal to formal	BLEURT	BLEU	ACC
Source	i say omarion.he has the hair clothes and body,a triple deal on one person.		-	
Reference 1	My choice is Omarion as he has high quality, hair, clothes, and body to create a triple deal in one person.		-	
Reference 2	I would say Omarion because he has the hair, clothes, and body; A triple deal on a single person.		-	
Reference 3	I pick Omarion, he has the hair, the clothes, and the body. A triple deal on one person.		-	
Reference 4	Omarion has the hair, clothes, and the body.		-	
PBMT-Combined (Rao and Tetreault, 2018)	I say omarion. he has the hair, clothes and body, the deal on one person.	-0.153	0.509	0.946
Bi-directional FT (Niu et al., 2018)	I say Omarion, he has the hair clothes and body, and a triple deal on one person.	-0.149	0.510	0.953
GPT-CAT (Wang et al., 2019)	I say Omarion. He has the hair, clothes, and body, a triple deal on one person.	0.044	0.585	**1.000**
S2S-SLS (Wang et al., 2020)	I say Omarion. He has the hair clothes and body, a triple deal on one person.	-0.035	0.350	**1.000**
Transformer (Zhang et al., 2020)	I say omarionhe has the hair clothes and body, a triple deal on one person.	-0.255	0.462	0.892
Chawla's (Chawla and Yang, 2020)	I say Marion because he has the hair, clothes and body, a triple deal on one person.	-0.538	0.534	0.989
OpenNMT	He has the hair clothes and body.	-0.540	0.139	0.998
OpenNMT with SC	I say Omarion, he has the hair clothes and body.	-0.389	0.558	0.969
OpenNMT with BLEU	I say Omarion. He has the hair clothes and body.	-0.325	0.147	**1.000**
OpenNMT with SC & BLEU	I say Omarion. He has the hair clothes and body.	-0.325	0.147	**1.000**
GPT-2 base	I say Omarion. He has the hair and body, a triple deal on one person.	-0.087	0.342	**1.000**
GPT-2 + SC	I say Omarion because he has the hair clothes and body.	-0.264	0.634	0.976
GPT-2 + BLEU	I say Omarion. He has the hair clothes and body, a triple deal on one person.	-0.035	0.350	**1.000**
GPT-2 + SC & BLEU	I say Omarion. He has the hair clothes and body, a triple deal on one person.	-0.035	0.350	**1.000**
BART base	I would say Omar. He has the hair, clothes, and body. It is a triple deal on one person.	-0.012	0.589	**1.000**
BART + SC	I would say Omar. He has the hair, clothes, and body. It is a triple deal on one person.	-0.012	0.589	**1.000**
BART + BLEU	I would say Omar. He has the hair, clothes, and body of a triple deal on one person.	-0.230	0.600	**1.000**
BART + SC & BLEU	I would say Omar. He has the hair, clothes, and body. It is a triple deal on one person.	-0.012	0.589	**1.000**
BART large + SC & BLEU	I would say Omarion. He has the hair, clothes, and body, a triple deal on one person.	**0.096**	**0.657**	**1.000**

System	From formal to informal	BLEURT	BLEU	ACC
Source	I suggest avoiding hot dogs, and not watching this movie with your little sister.		-	
Reference 1	Don't eat hot dogs, or watch this movie with your little sister!		-	
Reference 2	Don't do hot dogs or this movie with your kid sister.		-	
Reference 3	don't eat hot dogs and don't watch it w/ ur lil sis!		-	
Reference 4	Don't eat hot dogs or watch this flick with your lil sis!		-	
PBMT-Combined (Rao and Tetreault, 2018)	I suggest avoiding hot dogs, and not watching this movie with your little sister.	-0.298	0.417	0.004
Bi-directional FT (Niu et al., 2018)	I suggest avoiding hot dogs and not watching this movie with your little sister.	-0.233	0.437	0.009
OpenNMT	hott dogs and not watching this movie with ur little sister	-0.885	0.118	**1.000**
OpenNMT with SC	Im not watching this movie with your little sister...I suggest him hot dogs.	-0.765	0.349	0.981
OpenNMT with BLEU	Well, and not watching this movie with your little sister.	-0.826	0.445	0.633
OpenNMT with SC & BLEU	Can't watch this movie with your little sister.	-0.521	0.542	0.783
GPT-2 base	Don't watch this movie with your little sister.	-0.415	0.573	0.851
GPT-2 + SC	don't watch this movie with your little sister.	-0.415	0.599	**1.000**
GPT-2 + BLEU	Don't watch this movie with your little sister!	-0.360	0.634	0.919
GPT-2 + SC & BLEU	don't watch this movie with your little sister.	-0.415	0.599	**1.000**
BART base	avoid hot dogs and not watch this movie with your little sister.	**-0.016**	0.610	0.925
BART + SC	avoid hot dogs and not watch this movie with your little sister.	**-0.016**	0.610	0.925
BART + BLEU	avoid hot dogs and not watching this movie with your little sister.	-0.034	0.514	0.910
BART + SC & BLEU	avoid hot dogs and not watch this movie with your little sister.	**-0.016**	0.610	0.925
BART large + SC & BLEU	Avoid hot dogs and don't watch this movie with your little sister.	-0.171	**0.800**	0.825

Table A.3.1: Sample model outputs and their sentence-level scores on the E&M domain, where red denotes improperly generated words or content. Note that ACC indicates style confidence here.

Deep Context- and Relation-Aware Learning for Aspect-based Sentiment Analysis

Shinhyeok Oh[1*†], **Dongyub Lee**[2*], **Taesun Whang**[3], **Ilnam Park**[4], **Gaeun Seo**[4], **Eunggyun Kim**[4], **and Harksoo Kim**[5‡]
[1]Netmarble AI Center [2]Kakao Corp. [3]Wisenut Inc.
[4]Kakao Enterprise [5]Konkuk University

Abstract

Existing works for aspect-based sentiment analysis (ABSA) have adopted a unified approach, which allows the interactive relations among subtasks. However, we observe that these methods tend to predict polarities based on the literal meaning of aspect and opinion terms and mainly consider relations implicitly among subtasks at the word level. In addition, identifying multiple aspect–opinion pairs with their polarities is much more challenging. Therefore, a comprehensive understanding of contextual information w.r.t. the aspect and opinion are further required in ABSA. In this paper, we propose Deep Contextualized Relation-Aware Network (DCRAN), which allows interactive relations among subtasks with deep contextual information based on two modules (i.e., Aspect and Opinion Propagation and Explicit Self-Supervised Strategies). Especially, we design novel self-supervised strategies for ABSA, which have strengths in dealing with multiple aspects. Experimental results show that DCRAN significantly outperforms previous state-of-the-art methods by large margins on three widely used benchmarks.

1 Introduction

Aspect-based sentiment analysis (ABSA) is a task of identifying the sentiment polarity of associated aspect terms in a sentence. Generally, ABSA is composed of three subtasks, 1) aspect term extraction (ATE), 2) opinion term extraction (OTE), and 3) aspect-based sentiment classification (ASC). Given the sentence "*Food is good, but service is dreadful.*", ATE aims to identify two-aspect terms "*food*" and "*service*", and OTE aims to determine two-opinion terms "*good*" and "*dreadful*". Then, ASC assigns a sentiment polarity of each aspect: "*food* (*positive*)" and "*service* (*negative*)".

	Examples (Ground Truth)	Model	Aspect (Polarity)	Opinion
E1	I've had *better Japanese food* (neg) at a mall food court.	RACL	Japanese food (pos)	better
		DCRAN	Japanese food (neg)	better
E2	The *sushi* (neg) is cut in blocks *bigger* than my cell phone.	RACL	sushi (neu)	bigger
		DCRAN	sushi (neg)	bigger
E3	While the *smoothies* (neg) are a little *bigger* for me, the *fresh juices* (pos) are the *best* I have ever had !	RACL	smoothies (pos) fresh juices (pos)	bigger fresh best
		DCRAN	smoothies (neg) fresh juices (pos)	bigger best

Table 1: Examples of ABSA results comparing to previous approach (Chen and Qian, 2020) that we reimplement. All the results are based on BERT$_{base}$ model for a fair comparison. The polarity labels pos, neu, and neg, denote positive, neutral, and negative, respectively.

Existing works for ABSA have adopted a two-step approach, which considers each subtask separately (Tang et al., 2016; Xu et al., 2018). However, most recently, unified approaches have achieved significant performance improvements in ABSA task. Luo et al. (2020) focused on modeling the interactions between aspect terms and Chen and Qian (2020) exploited dyadic and triadic relations between subtasks (i.e., ATE, OTE, ASC).

Despite the impressive results, their methods have two limitations. First, they only consider relations among subtasks at the word level and do not explicitly utilize contextualized information of the whole sequence. For example, E1 in Table 1, the opinion term "*better*" seems to represent positive opinion of "*Japanese food*". However, the authentic meaning of E1 is "*The Japanese food I have had at the food court was more delicious than the one I had at this restaurant*". Thus, previous approaches tend to assign polarities based on the literal meaning of aspect and opinion terms (E2). Second, identifying multiple aspect–opinion pairs and their polarities is much more challenging as the model needs to not only detect multiple aspects and

[*]These two authors equally contributed to this work.
[†]This work was done while the author was an intern at Kakao Enterprise.
[‡]Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 495–503
August 1–6, 2021. ©2021 Association for Computational Linguistics

opinions but also correctly predict each polarity of the aspect (E3).

To address the aforementioned issues, we propose Deep Contextualized Relation-Aware Network (DCRAN) for ABSA. DCRAN not only implicitly allows interactive relations among the subtasks of ABSA, but also explicitly considers their relations by using contextual information. Our main contributions are as follows: 1) We design aspect and opinion propagation decoder so that the model has a comprehensive understanding of the whole context, and thus it results in better prediction of the polarity. 2) We propose novel self-supervised strategies for ABSA, which are highly effective in dealing with multiple aspects and considering deep contextualized information with the aspect and opinion terms. To the best of our knowledge, it is the first attempt to design explicit self-supervised methods for ABSA. 3) Experimental results demonstrate that DCRAN significantly outperforms previous state-of-the-art methods on three widely used benchmarks.

2 DCRAN: Deep Contextualized Relation-Aware Network

2.1 Task Definition

Given a sentence $S = \{w_1, w_2, ..., w_n\}$, where n denotes the number of tokens, we aim to solve three subtasks: aspect term extraction (ATE), opinion term extraction (OTE), and aspect-based sentiment classification (ASC) as sequence labeling problems. ATE task aims to identify a sequence of aspect terms $Y^a = \{y_1^a, y_2^a, ..., y_n^a\}$, where $y_i^a \in \{B, I, O\}$, and OTE task aims to identify a sequence of opinion terms $Y^o = \{y_1^o, y_2^o, ..., y_n^o\}$, where $y_i^o \in \{B, I, O\}$ of aspect and opinion terms, respectively. Likewise, ASC task aims to assign a sequence of polarities $Y^p = \{y_1^p, y_2^p, ..., y_n^p\}$, where $y_i^p \in \{POS, NEU, NEG, O\}$. The labels *POS*, *NEU*, and *NEG* denote *positive*, *neutral*, and *negative*, respectively.

2.2 Task-Shared Representation Learning

Following existing works, we utilize pre-trained language models, such as BERT (Devlin et al., 2019) and ELECTRA (Clark et al., 2020) as the shared encoder to construct context representation, which is shared by subtasks: ATE, OTE, and ASC. Given a sentence $S = \{w_1, w_2, ..., w_n\}$, pre-trained language models take the input sequence, $\mathbf{X}_{\text{absa}} = [[\text{CLS}]\, w_1\, w_2 \ldots w_n\, [\text{SEP}]]$, and output a sequence of the shared context representation, $H = \{h_{[\text{CLS}]}, h_1, h_2, ..., h_n, h_{[\text{SEP}]}\} \in \mathbb{R}^{d_h \times (n+2)}$, where d_h represents a dimension of the shared encoder. We represent the parameters of the shared encoder as Θ_s. Then, we utilize a single-layer feed-forward neural network (FFNN) as,

$$\begin{aligned} Z^a &= (W_1 h_{[1:n+1]} + b_1) \\ \hat{Y}^a &= \text{softmax}(W_2 Z^a + b_2), \end{aligned} \tag{1}$$

where $W_1 \in \mathbb{R}^{d_h \times d_h}$ and $W_2 \in \mathbb{R}^{3 \times d_h}$ are trainable parameters. The parameters of a single-layer FFNN are represented as Θ_a for aspect term extraction. The objective of aspect term extraction is minimizing the negative log-likelihood (NLL) loss: $\mathcal{L}_{\text{ate}}(\Theta_s, \Theta_a) = -\sum \log p(Y^a|H)$. Likewise, Z^o and $\hat{Y}^o$ are obtained as in Equation 1. Then, the NLL loss of opinion term extraction is defined as, $\mathcal{L}_{\text{ote}}(\Theta_s, \Theta_o) = -\sum \log p(Y^o|H)$.

2.3 Aspect and Opinion Propagation

We utilize the transformer-decoder (Vaswani et al., 2017) to consider relations of aspect and opinion while predicting a sequence of polarities. Our transformer-decoder is mainly composed of a multi-head self-attention, two multi-head cross attention, and a feed-forward layer. The multi-head self-attention takes shared context representation H as,

$$U^h = \text{LN}(H + \text{SelfAttn}(H, H, H)) \tag{2}$$

and U^h, Z^a, and Z^o are fed into two steps of cross multi-head attention as,

$$U^a = \text{LN}(U^h + \text{CrossAttn}(U^h, Z^a, Z^a)) \tag{3}$$

$$U^o = \text{LN}(U^a + \text{CrossAttn}(U^a, Z^o, Z^o)) \tag{4}$$

where LN represents layer norm (Ba et al., 2016). Note that Equation 3 and 4 represent aspect and opinion propagation, respectively. Then U^o is fed into a single-layer FFNN to obtain a sequence of polarities Y^p. The objective of aspect-based sentiment analysis is minimizing the NLL loss: $\mathcal{L}_{\text{asc}}(\Theta_s, \Theta_a, \Theta_o, \Theta_p) = -\sum \log p(Y^p|H, Z^a, Z^o)$. The architecture of the aspect and opinion propagation is described in Figure 1-(a).

2.4 Explicit Self-Supervised Strategies

To further exploit the aspect–opinion relation with contextualized information of a sentence, we propose explicit self-supervised strategies consisting of two auxiliary tasks: 1) type-specific masked term

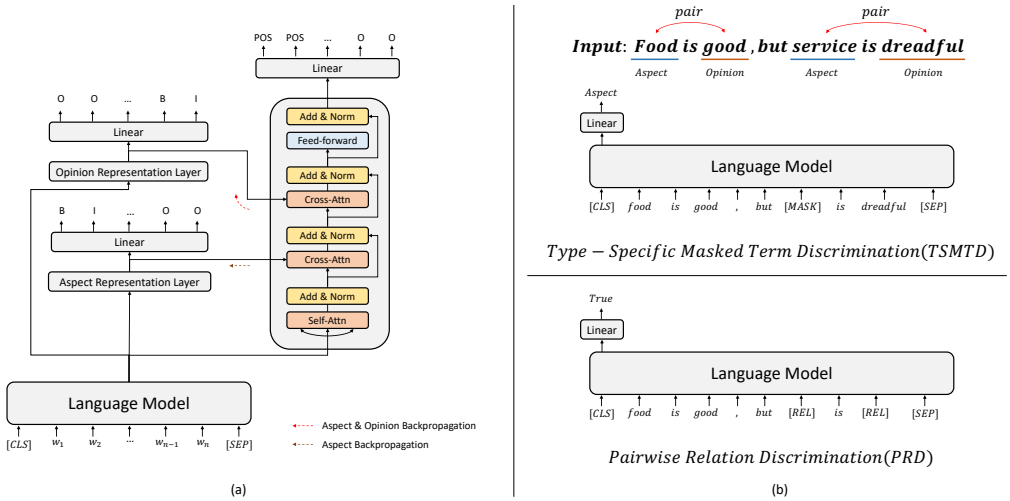


Figure 1: Overall architecture of Deep Contextualized Relation-Aware Network (DCRAN) for ABSA.

discrimination (TSMTD) and 2) pairwise relations discrimination (PRD). The examples of *Explicit Self-Supervised Strategies* are described in Figure 1-(b).

Type-Specific Masked Term Discrimination In the type-specific masked term discrimination task, we uniformly mask aspects, opinions, and terms that do not correspond to both, using the special token [MASK]. The input sequence of a masked sentence is represented as, $\mathbf{X}_{\text{tsmtd}} = [[\text{CLS}]\, w_1 \ldots [\text{MASK}]_i \ldots w_n\, [\text{SEP}]]$, and is fed into pre-trained language models. Then, the output representation of [CLS] token is used to classify which type of term is masked in a sentence as,

$$\hat{Y}^m = \text{softmax}(W_3 h_{[\text{CLS}]} + b_3),$$

where $W_3 \in \mathbb{R}^{3\times d_h}$ represents trainable parameters and $\hat{Y}^m \in \{Aspect, Opinion, O\}$. The parameters of a linear projection layer are represented as Θ_m for the type-specific masked term discrimination. Then, the NLL loss of the type-specific masked term discrimination is defined as: $\mathcal{L}_{\text{tsmtd}}(\Theta_s, \Theta_m) = -\sum \log p(Y^m|H)$. This allows the model to explicitly exploit sentence information by discriminating what kind of term is masked.

Pairwise Relations Discrimination In this task, we uniformly replace both aspects and opinion terms using the special token [REL]. The input sequence of a masked sentence is represented as, $\mathbf{X}_{\text{prd}} = [[\text{CLS}]\, w_1 \ldots [\text{REL}]_i \ldots [\text{REL}]_j \ldots w_n\, [\text{SEP}]]$, and is fed into pre-trained language models. Then, the output representation of [CLS] token is used to discriminate whether the replaced tokens have a pairwise relation as,

$$\hat{Y}^r = \text{softmax}(W_4 h_{[\text{CLS}]} + b_4),$$

where $W_4 \in \mathbb{R}^{2\times d_h}$ represents trainable parameters and $\hat{Y}^r \in \{True, False\}$. The parameters of a linear projection layer are represented as Θ_r for the pairwise relations discrimination. Then, the NLL loss of the pairwise relations discrimination is defined as: $\mathcal{L}_{\text{prd}}(\Theta_s, \Theta_r) = -\sum \log p(Y^r|H)$. We describe the negative sampling method to replace aspects and opinion terms in Appendix A.2.

2.5 Joint Learning Procedure

All these tasks are jointly trained, and the final objective is defined as,

$$\mathcal{L}_{\text{absa}} = \mathcal{L}_{\text{ate}} + \mathcal{L}_{\text{ote}} + \mathcal{L}_{\text{asc}}$$
$$\mathcal{L}_{\text{aux}} = \mathcal{L}_{\text{tsmtd}} + \mathcal{L}_{\text{prd}}$$
$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{absa}} + \alpha \mathcal{L}_{\text{aux}}$$

where α is a hyper-parameter determining the degree of auxiliary tasks. Note that the parameters Θ_s are optimized for all subtasks. Especially, the parameters Θ_s are further optimized through $\mathcal{L}_{\text{tsmtd}}$ and $\mathcal{L}_{\text{prd}}$ to explicitly exploit the relations between aspect and opinion with context meaning.

3 Experiments

3.1 Experimental Setup

We evaluate our model on three widely used sentiment analysis benchmarks: laptop reviews (LAP14), restaurant reviews (REST14) from (Pontiki et al., 2014), and restaurant reviews (REST15)

		LAP14				REST14				REST15			
		ATE-F1	OTE-F1	ASC-F1	ABSA-F1	ATE-F1	OTE-F1	ASC-F1	ABSA-F1	ATE-F1	OTE-F1	ASC-F1	ABSA-F1
MNN (Wang et al., 2018)	GloVe	76.94	77.77	65.98	53.80	83.05	84.55	68.45	63.87	70.24	69.38	57.90	56.57
E2E-TBSA (Li et al., 2019)	GloVe	77.34	76.62	68.24	55.88	83.92	84.97	68.38	66.60	69.40	71.43	58.81	57.38
DOER (Luo et al., 2019)	GloVe	80.21	-	60.18	56.71	84.63	-	64.50	68.55	67.47	-	36.76	50.31
IMN^{-d} (He et al., 2019)	GloVe	78.46	78.14	69.62	57.66	84.01	85.64	71.90	68.32	69.80	72.11	60.65	57.91
RACL (Chen and Qian, 2020)	GloVe	81.99	79.76	71.09	60.63	85.37	85.32	74.46	70.67	72.82	78.06	68.69	60.31
WHW (Peng et al., 2020)	GloVe	-	74.84	-	62.34	-	82.45	-	71.95	-	78.02	-	65.79
IKTN (Liang et al., 2020)	BERT$_{base}$	80.89	78.90	73.42	62.34	86.13	86.62	74.35	71.75	71.63	76.79	69.85	62.33
SPAN (Hu et al., 2019)	BERT$_{large}$	82.34	-	62.50	61.25	86.71	-	71.75	73.68	74.63	-	50.28	62.29
IMN^{-d} (He et al., 2019)	BERT$_{large}$	77.55	**81.00**	75.56	61.73	84.06	85.10	75.67	70.72	69.90	73.29	70.10	60.22
Dual-MRC (Mao et al., 2021)	BERT$_{large}$	82.51	-	75.97	65.94	86.60	-	**82.04**	75.95	**75.08**	-	73.59	65.08
RACL (Chen and Qian, 2020)	BERT$_{large}$	81.79	79.72	73.91	63.40	86.38	**87.18**	81.61	75.42	73.99	76.00	74.91	66.05
DCRAN (Ours)	BERT$_{base}$	81.76	78.84	77.02	65.18	88.21	86.36	78.67	75.77	71.61	75.86	73.30	63.19
	BERT$_{large}$	**83.40**	79.72	**78.75**	**68.07**	**88.73**	86.07	80.64	**77.28**	74.45	**78.45**	**76.30**	**67.92**
	ELECTRA$_{base}$	**85.69**	**80.19**	79.36	70.22	89.64	87.30	84.12	80.00	77.41	78.80	**78.55**	71.67
	ELECTRA$_{large}$	85.61	79.77	**80.78**	**71.47**	**89.67**	**87.59**	**84.22**	**80.32**	**79.68**	**79.90**	77.99	**73.67**

Table 2: Evaluation results on the LAP14, REST14, and REST15 datasets, which are provided by Chen and Qian (2020). All the results except ours are cited from the existing works (Chen and Qian, 2020; Peng et al., 2020; Mao et al., 2021) and all the baselines are described in Appendix A.4. We report average results over five runs with random initialization. The best scores are in bold, and the second-best scores are underlined depending on the types of the pre-trained language model. '-' denotes unreported results.

from (Pontiki et al., 2015). Primitive versions of these benchmarks only provide aspect terms and sentiment polarities, while opinion terms are provided by Wang et al. (2016, 2017) later. Recently, Fan et al. (2019) provides aspect-opinion pairwise datasets (Section 2.4). Following He et al. (2019), we set four evaluation metrics: ATE-F1, OTE-F1, ASC-F1, and ABSA-F1. The ATE-F1, OTE-F1, and ASC-F1 measure each subtask's F-1 scores, and ABSA-F1 measures complete ABSA, which counts only when both ATE and ASC predictions are correct.

3.2 Quantitative Results

Table 2 reports the quantitative results on the LAP14, REST14, and REST15 datasets. Our experiments utilize two pre-trained language models such as BERT and ELECTRA, for the shared encoder. First, we observe that DCRAN-BERT$_{base}$ shows slightly lower ABSA-F1 scores than previous state-of-the-art methods, which is based on BERT$_{large}$, on the REST14 and LAP14 datasets except for the REST15 dataset. This suggests that our proposed methods are highly effective for ABSA. Overall, DCRAN-BERT$_{large}$ significantly outperforms previous state-of-the-art methods in all metrics. Another observation is that ELECTRA based models outperform BERT based models. As a result, DCRAN-ELECTRA$_{large}$ achieves absolute gains over previous state-of-the-art results by 5.5%, 4.4%, and 7.6% in ABSA-F1 on the LAP14, REST14, and REST15 datasets, respectively.

3.3 Ablation Study

To study the effectiveness of the aspect propagation (AP), opinion propagation (OP), type-specific masked term discrimination (TSMTD), and pairwise relations discrimination (PRD), we conduct ablation experiments on the REST14 dataset. We set the baseline model that did not utilize aspect and opinion propagation and explicit self-supervised strategies. When the AP and OP are not utilized, a single-layer FFNN is utilized as in Equation 1 to predict a sequence of polarities Y^p instead of transformer-decoder. As shown in Table 3, we can observe that the AP is more effective than the OP, and scores drop significantly when not utilizing the AP and OP. In the case of explicit self-supervised strategies, we can observe that the PRD is more effective than the TSMTD. As the PRD objective is discriminating whether the replace tokens have a pairwise aspect–opinion relations, it allows the model to more exploit the relations between aspect and opinion at a sentence level.

		ABSA-F1
	DCRAN-ELECTRA$_{base}$	**80.00**†
Aspect and Opinion Propagation	w/o AP	79.44†
	w/o OP	79.58†
	w/o AP & OP	79.08†
Explicit Self-Supervised Strategies	w/o TSMTD	79.56†
	w/o PRD	79.40†
	w/o TSMTD & PRD	79.03†
Baseline	w/o & AP & OP & TSMTD & PRD	78.61

Table 3: Ablation study on the REST14 dataset. We choose DCRAN-ELECTRA$_{base}$ as the baseline. † denotes statistical significance (p-value < 0.05).

3.4 Aspect Analysis

We conduct aspect analysis by comparing sentences with single- and multiple-aspect. As shown in Table 4, *Aspect and Opinion Propagation* signif-

		REST14		REST15	
		ABSA-F1	Sent-level Acc.	ABSA-F1	Sent-level Acc.
Single-Aspect	DCRAN_ELECTRA$_{base}$	**78.62**	**74.48**	**66.23**	**67.69**
	w/o TSMTD & PRD	78.42	73.79	64.21	66.67
	w/o TSMTD & PRD & AP & OP	77.45	73.10	62.50	64.29
Multiple-Aspect	DCRAN_ELECTRA$_{base}$	**81.19**	**64.24**	**68.20**	**52.34**
	w/o TSMTD & PRD	80.22	61.70	65.16	48.60
	w/o TSMTD & PRD & AP & OP	79.88	61.39	64.84	46.73

Table 4: Aspect analysis on the REST14 and REST15 datasets. Comparisons of ABSA-F1 and sentence-level accuracy results for the case when the sentence contains single-aspect or multiple-aspect.

icantly improves performance when the sentence contains a single-aspect, while a small increase is observed w.r.t. the case of multiple-aspect. Although considering the relations between aspect and opinion implicitly can improve performance w.r.t. the case of single-aspect, it is not sufficient for inducing performance improvement for the multiple-aspect case. It suggests that additional explicit tasks are further required to identify multiple-aspect with corresponding opinions, which helps the model assign polarities correctly. In the case of multiple-aspect, *Explicit Self-Supervised Strategies* show absolute ABSA-F1 improvements of 0.97% (80.22% $\rightarrow$ 81.19%) and 3.04% (65.16% $\rightarrow$ 68.20) on the REST14 and REST15 datasets, respectively. This indicates explicit self-supervised strategies are highly effective for correctly identifying ABSA when the sentence contains multiple-aspect. In addition, the performance gain by *Explicit Self-Supervised Strategies* in Table 3 is mostly derived from the multiple-aspect cases (+0.97%), thus our proposed model has strengths in dealing with multiple aspects.

In ABSA, it is important to accurately predict all aspects and corresponding sentiment polarities in one sentence. Since ABSA-F1 is a word-level based metric, it still has a limitation to evaluate whether all aspects and corresponding polarities are correct or not. Therefore, we also evaluate our method with sentence-level accuracy; the number of sentences that accurately predicted all aspects and polarity in a sentence divided by total number of sentences. Unlike ABSA-F1, the sentence-level accuracy of multiple-aspect is lower than that of single-aspect, which implies identifying multiple aspects and their polarities is more challenging. In the case of multiple-aspect, our *Explicit Self-Supervised Strategies* leads significant sentence-level accuracy improvements of 2.54% (61.70% $\rightarrow$ 64.24%) and 3.74% (48.60% $\rightarrow$ 52.34%) on the REST14 and REST15 datasets, respectively. However, we observe only small improvements in sentence-level accuracy on both datasets when the sentence contains single-aspect. From these observations, we demonstrate that our proposed method is highly effective for the case when the sentence contains multiple aspects.

4 Conclusion

In this paper, we proposed the Deep Contextualized Relation-Aware Network (DCRAN) for aspect-based sentiment analysis. DCRAN allows interaction between subtasks implicitly in a more effective manner and two explicit self-supervised strategies for deep context- and relation-aware learning. We obtained the new state-of-the-art results on three widely used benchmarks.

Acknowledgements

We thank the anonymous reviewers, Dongsuk Oh, Jungwoo Lim, and Heuiseok Lim for their constructive comments. This work was partially supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(No.2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1F1A1069737).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 547–556.

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 3685–3694.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2509–2518.

Aitor García-Pablos, Montse Cuadros, and German Rigau. 2018. W2vlda: almost unsupervised system for aspect based sentiment analysis. *Expert Systems with Applications*, 91:127–137.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515.

Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. 2021. Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 537–546.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6714–6721.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4194–4200.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2020. An iterative knowledge transfer network with routing for aspect-based sentiment analysis. *arXiv preprint arXiv:2004.01935*.

Huaishao Luo, Lei Ji, Tianrui Li, Daxin Jiang, and Nan Duan. 2020. Grace: Gradient harmonized and cascaded labeling for aspect-based sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 54–64.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: Dual cross-shared RNN for aspect term-polarity co-extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 591–601.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4737–4742.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8600–8607.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Feixiang Wang, Man Lan, and Wenting Wang. 2018. Towards a one-stop solution to both aspect extraction and sentiment analysis tasks with neural multi-task learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, page 3316–3322.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and CNN-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598.

Jianfei Yu, Jing Jiang, and Rui Xia. 2018. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):168–177.

A Appendix

A.1 Related Work

Existing works have studied a two-step approach for ABSA. In a two-step approach, each model for ATE, OTE, and ASC are separately trained and are merged in a pipelined manner (Wang et al., 2016; Tang et al., 2016; Wang et al., 2017; He et al., 2017; Xu et al., 2018; Yu et al., 2018; Li et al., 2018; Chen and Qian, 2019). However, the errors from other tasks can be propagated to the ASC and can degrade performance after all.

Most recently, a unified approach that comprised of joint approach (García-Pablos et al., 2018; Luo et al., 2019; He et al., 2019; Luo et al., 2020) and collapsed approach (Li and Lu, 2017; Ma et al., 2018; Wang et al., 2018; Li et al., 2019) has been proposed. A joint approach labels each word with different tag sets for each task: ATE, OTE, and ASC. On the other hand, a collapsed approach labels each word as the combined one of ATE and ASC, such as "B-POSITIVE" and "I-POSITIVE", where "B" and "I" represent the aspect term boundary, and "POSITIVE" represents polarity. However, in a collapsed approach, the relations among subtasks cannot be effectively exploited because subtasks need to share all representation without distinction of each task. Therefore, a joint training approach allows the interactive relations between subtasks, while a collapsed approach is not.

A.2 Negative Sampling Algorithm for Pairwise Relations Discrimination

Algorithm 1 describes the negative sampling procedure in pairwise relations discrimination. The get_sample function takes a list of aspect-opinion pairs in a sentence and replaces them with [REL] tokens. Then, if the replaced tokens have pairwise relations, set the target label as True, and set as False if not. The get_pair function randomly selects a pairwise aspect and opinion, and the get_neg_pair function selects aspects and opinions of different pairs when there are two or more pairs in a sentence.

Algorithm 1 Negative Sampling Algorithm for Pairwise Relations Discrimination

Input: $pairs$: list of aspect–opinion pairs in a sentence
Output: $pair, target$
  function GET_SAMPLE($pairs$)
    if count($pairs$) == 0 **then**
      return None, None
    else if count($pairs$) == 1 **then**
      return $pairs$[0], True
    else
      $random$ = $\{0 < random \leq 1\}$
      if $random \leq 0.25$ **then**
        return get_pair($pairs$), True
      else
        return get_neg_pair($pairs$), False

A.3 Implementation Details

We implemented our model by using the PyTorch (Paszke et al., 2019) deep learning library based on the open source[1] (i.e., Transformers (Wolf et al., 2020)). For the shared encoder, we adopt four types of pre-trained language models: BERT$_{base}$, BERT$_{large}$, ELECTRA$_{base}$, and ELECTRA$_{large}$. We set the batch size to 64 for the $base$ model, 12 for the BERT$_{large}$ and 32 for the ELECTRA$_{large}$. We set the initial learning rate to 5e-5 for BERT$_{base}$ and ELECTRA$_{base}$, 2e-5 for BERT$_{large}$, and 5e-6 for ELECTRA$_{large}$. For the transformer decoder, we set the number of heads in multi-head attention and hidden layers to 2 among range from 2 to 6, and hidden dimension size to 768. In the case of α, we obtained the best results when α is 1. The average runtime for each approach was about 20 seconds for BERT$_{base}$ and ELECTRA$_{base}$, and 90 seconds for BERT$_{large}$ and ELECTRA$_{large}$. We train our models using AdamP (Heo et al., 2021) optimizer and conduct experiments with Tesla V100 GPU for all the experiments.

A.4 Baselines

We compare our model with the following previous works[2].

MNN (Wang et al., 2018) is a multi-task model for ATE and ASC using attention mechanisms to learn the joint representation of aspect and polarity relations.

E2E-TBSA (Li et al., 2019) is an end-to-end model of the collapsed approach for ATE and ASC. Additionally, it introduces the auxiliary OTE task without explicit interaction.

[1] https://github.com/huggingface/transformers

[2] We do not compare our work with GRACE (Luo et al., 2020) as Luo et al. (2020) contains *conflict* tag in polarities.

	Examples (Ground Truth)	Model	Aspect (Polarity)	Opinion
E1	I have worked in restaurants and cook a lot, and there is no way a maggot should be able to get into *well prepared food* (neg).	RACL	food (pos)	well
		DCRAN w/o	food (pos)	well prepared
		DCRAN	food (neg)	well prepared
E2	All in all, I would return - as it was a *beautiful restaurant* (pos) - but I hope the *staff* (neg) pays more attention to the little details in the future.	RACL	-	-
		DCRAN w/o	restaurant (pos) staff (pos)	beautiful
		DCRAN	restaurant (pos) staff (neg)	beautiful
E3	I have never been so *disgusted* by both *food* (neg) and *service* (neg)	RACL	food (pos) service (pos)	disgusted
		DCRAN w/o	food (pos) service (neg)	disgusted
		DCRAN	food (neg) service (neg)	disgusted

Table 5: Case study on the REST15 dataset. Model comparison between previous state-of-the-art method (RACL) (Chen and Qian, 2020) and our proposed method (DCRAN). DCRAN w/o denotes DCRAN without *Explicit Self-Supervised Strategies* (Section 2.4). All models are built based on the BERT$_{base}$ model. The polarity labels pos, neu, and neg denote positive, neutral, and negative, respectively. '-' denotes that the model failed to extract corresponding terms.

DOER (Luo et al., 2019) is a dual cross-shared RNN framework that jointly trains ATE and ASC. It considers relations between aspect and polarity.

IMN (He et al., 2019) is a multi-task model for ATE and ASC with separate labels. The OTE task is fused into ATE by constructing five-class labels.

WHW (Peng et al., 2020) is a unified two-stage framework to extract (aspect, opinion, polarity) triples as a result of ATE, OTE, and ASC.

IKTN (Liang et al., 2020) is an iterative knowledge transfer network for ABSA considering the semantic correlations among the ATE, OTE, and ASC.

SPAN (Hu et al., 2019) is a pipeline approach to solve ATE and ASC using BERT$_{large}$. It uses a multi-target extractor for ATE and a polarity classifier for ASC.

RACL (Chen and Qian, 2020) defines interactive relations among ATE, OTE, and ASC. It proposes relation propagation mechanisms through the stacked multi-layer network.

Dual-MRC (Mao et al., 2021) leverages two machine reading comprehension problems to solve ATE and ASC. It jointly trains two BERT-MRC models sharing parameters.

A.5 Case Study

In E1 and E3, while all models correctly extract both aspect and opinion, RACL and DCRAN w/o make inaccurate polarities predictions based on the words having superficial meaning (i.e., *well prepared*, *disgusted*). Especially, E3 expresses a sarcastic opinion about aspect terms throughout the sentence. It suggests that these models cannot understand the authentic meaning of the sentence. On the other hand, DCRAN grasps the entire context and predicts the correct polarity corresponding to its aspect. In E2, the evidence for understanding the actual meaning of the aspect term *staff* is not specified in a word-level opinion and expressed in a sentence like "*I hope the staff pays more attention to the little details in the future*". In this case, RACL can not extract aspect and opinion terms, and DCRAN w/o make inaccurate polarities predictions for the aspect term *staff* based on the opinion term *beautiful*. However, DCRAN with *Explicit Self-Supervised Strategies* understands the sentence expressing an opinion on the *staff* and predicts correctly.

Towards Generative Aspect-Based Sentiment Analysis*

Wenxuan Zhang[1], Xin Li[2], Yang Deng[1], Lidong Bing[2] and Wai Lam[1]
[1]The Chinese University of Hong Kong
[2]DAMO Academy, Alibaba Group
{wxzhang,ydeng,wlam}@se.cuhk.edu.hk
{xinting.lx,l.bing}@alibaba-inc.com

Abstract

Aspect-based sentiment analysis (ABSA) has received increasing attention recently. Most existing work tackles ABSA in a discriminative manner, designing various task-specific classification networks for the prediction. Despite their effectiveness, these methods ignore the rich label semantics in ABSA problems and require extensive task-specific designs. In this paper, we propose to tackle various ABSA tasks in a unified generative framework. Two types of paradigms, namely annotation-style and extraction-style modeling, are designed to enable the training process by formulating each ABSA task as a text generation problem. We conduct experiments on four ABSA tasks across multiple benchmark datasets where our proposed generative approach achieves new state-of-the-art results in almost all cases. This also validates the strong generality of the proposed framework which can be easily adapted to arbitrary ABSA task without additional task-specific model design.[1]

1 Introduction

Aspect-based sentiment analysis (ABSA), aiming at mining fine-grained opinion information towards specific aspects, has attracted increasing attention in recent years (Liu, 2012). Multiple fundamental sentiment elements are involved in ABSA, including the aspect term, opinion term, aspect category, and sentiment polarity. Given a simple example sentence "*The pizza is delicious.*", the corresponding elements are "*pizza*", "*delicious*", "*food quality*" and "*positive*", respectively.

The main research line of ABSA focuses on the identification of those sentiment elements such as extracting the aspect term (Liu et al., 2015; Yin et al., 2016; Li et al., 2018; Ma et al., 2019) or classifying the sentiment polarity for a given aspect (Wang et al., 2016; Chen et al., 2017; Jiang et al., 2019; Zhang and Qian, 2020). To provide more detailed information, many recent studies propose to jointly predict multiple elements simultaneously (Li et al., 2019a; Wan et al., 2020; Peng et al., 2020; Zhao et al., 2020). Taking the Unified ABSA (UABSA, also called End-to-End ABSA) task as an example, it tries to simultaneously predict the mentioned aspect terms and the corresponding sentiment polarities (Luo et al., 2019; He et al., 2019).

In general, most ABSA tasks are formulated as either sequence-level or token-level classification problems (Li et al., 2019b). By designing task-specific classification networks, the prediction is made in a discriminative manner, using the class index as labels for training (Huang and Carley, 2018; Wan et al., 2020). However, these methods ignore the label semantics, *i.e.*, the meaning of the natural language labels, during the training process. Intuitively, knowing the meaning of "*food quality*" and "*restaurant ambiance*", it can be much easier to identify that the former one is more likely to be the correct aspect category for the concerned aspect "*pizza*". Such semantics of the label can be more helpful for the joint extraction of multiple sentiment elements, due to the complicated interactions of those involved elements. For example, understanding "*delicious*" is an adjective for describing the food such as "*pizza*" could better lead to the prediction of aspect opinion pair ("*pizza*", "*delicious*"). Another issue is that different classification models are proposed to suit the need of different ABSA problems, making it difficult to adapt the model from one to another.

Motivated by recent success in formulating sev-

* Work done when Wenxuan Zhang was an intern at Alibaba. The work described in this paper is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719).

[1]The data and code can be found at https://github.com/IsakZhang/Generative-ABSA

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 504–510
August 1–6, 2021. ©2021 Association for Computational Linguistics

eral language understanding problems such as named entity recognition, question answering, and text classification as generation tasks (Raffel et al., 2020; Athiwaratkun et al., 2020), we propose to tackle various ABSA problems in a unified generative approach in this paper. It can fully utilize the rich label semantics by encoding the natural language label into the target output. Moreover, this unified generative model can be seamlessly adapted to multiple tasks without introducing additional task-specific model designs.

In order to enable the **G**enerative **A**spect-based **S**entiment analysis (GAS), we tailor-make two paradigms, namely annotation-style and extraction-style modeling to transform the original task as a generation problem. Given a sentence, the former one adds annotations on it to include the label information when constructing the target sentence; while the latter directly adopts the desired natural language label of the input sentence as the target. The original sentence and the target sentence produced by either paradigm can then be paired as a training instance of the generation model. Furthermore, we propose a prediction normalization strategy to handle the issue that the generated sentiment element falls out of its corresponding label vocabulary set. We investigate four ABSA tasks including Aspect Opinion Pair Extraction (AOPE), Unified ABSA (UABSA), Aspect Sentiment Triplet Extraction (ASTE), and Target Aspect Sentiment Detection (TASD) with the proposed unified GAS framework to verify its effectiveness and generality.

Our main contributions are 1) We tackle various ABSA tasks in a novel generative manner; 2) We propose two paradigms to formulate each task as a generation problem and a prediction normalization strategy to refine the generated outputs; 3) We conduct experiments on multiple benchmark datasets across four ABSA tasks and our approach surpasses previous state-of-the-art in almost all cases. Specifically, we obtain 7.6 and 3.7 averaged gains on the challenging ASTE and TASD task respectively.

2 Generative ABSA (GAS)

2.1 ABSA with Generative Paradigm

In this section, we describe the investigated ABSA tasks and the proposed two paradigms, namely, annotation-style and extraction-style modeling.

Aspect Opinion Pair Extraction (AOPE) aims to extract aspect terms and their corresponding opinion terms as pairs (Zhao et al., 2020; Chen et al., 2020). Here is an illustrative example of our generative formulations for the AOPE task:

`Input`: Salads were fantastic, our server was also very helpful.
`Target (Annotation-style)`: [Salads | fantastic] were fantastic here, our [server | helpful] was also very helpful.
`Target (Extraction-style)`: (Salads, fantastic); (server, helpful)

In the annotation-style paradigm, to indicate the pair relations between the aspect and opinion terms, we append the associated opinion modifier to each aspect term in the form of [*aspect* | *opinion*] for constructing the target sentence, as shown in the above example. The prediction of the coupled aspect and opinion term is thus achieved by including them in the same bracket. For the extraction-style paradigm, we treat the desired pairs as the target, which resembles direct extraction of the expected sentiment elements but in a generative manner.

Unified ABSA (UABSA) is the task of extracting aspect terms and predicting their sentiment polarities at the same time (Li et al., 2019a; Chen and Qian, 2020). We also formulate it as an (*aspect*, *sentiment polarity*) pair extraction problem. For the same example given above, we aim to extract two pairs: (*Salads*, *positive*) and (*server*, *positive*). Similarly, we replace each aspect term as [*aspect* | *sentiment polarity*] under the annotation-style formulation and treat the desired pairs as the target output in the extraction-style paradigm to reformulate the UABSA task as a text generation problem.

Aspect Sentiment Triplet Extraction (ASTE) aims to discover more complicated (*aspect*, *opinion*, *sentiment polarity*) triplets (Peng et al., 2020):

`Input`: The Unibody construction is solid, sleek and beautiful.
`Target (Annotation-style)`: The [Unibody construction | positive | solid, sleek, beautiful] is solid, sleek and beautiful.
`Target (Extraction-style)`: (Unibody construction, solid, positive); (Unibody construction, sleek, positive); (Unibody construction, beautiful, positive);

As shown above, we annotate each aspect term with its corresponding sentiment triplet wrapped in the bracket, *i.e.*, [*aspect*|*opinion*|*sentiment polarity*] for the annotation-style modeling. Note that

we will include all the opinion modifiers of the same aspect term within the same bracket to predict the sentiment polarities more accurately. For the extraction-style paradigm, we just concatenate all triplets as the target output.

Target Aspect Sentiment Detection (TASD) is the task to detect all (*aspect term*, *aspect category*, *sentiment polarity*) triplets for a given sentence (Wan et al., 2020), where the aspect category belongs to a pre-defined category set. For example,

> `Input`: A big disappointment, all around. The pizza was cold and the cheese wasn't even fully melted.
> `Target (Annotation-style)`: A big disappointment, all around. The [pizza | food quality | negative] was cold and the [cheese | food quality | negative] wasn't even fully melted [null | restaurant general | negative].
> `Target (Extraction-style)`: (pizza, food quality, negative); (cheese, food quality, negative); (null, restaurant general, negative);

Similarly, we pack each aspect term, the aspect category it belongs to, and its sentiment polarity into a bracket to build the target sentence for the annotation-style method. Note that we use a bigram expression for the aspect category instead of the original uppercase form "FOOD#QUALITY" to make the annotated target sentence more natural. As presented in the example, some triplets may not have explicitly-mentioned aspect terms, we thus use "*null*" to represent it and put such triplets at the end of the target output. For the extraction-style paradigm, we concatenate all the desired triplets, including those with implicit aspect terms, as the target sentence for sequence-to-sequence learning.

2.2 Generation Model

Given the input sentence $\boldsymbol{x}$, we generate a target sequence $\boldsymbol{y}'$, which is either based on the annotation-style or extraction-style paradigm as described in the last section, with a text generation model $f(\cdot)$. Then the desired sentiment pairs or triplets $\boldsymbol{s}$ can be decoded from the generated sequence $\boldsymbol{y}'$. Specifically, for the annotation-style modeling, we extract the contents included in the bracket "[]" from $\boldsymbol{y}'$, and separate different sentiment elements with the vertical bar "|". If such decoding fails, *e.g.*, we cannot find any bracket in the output sentence or the number of vertical bars is not as expected, we ignore such predictions. For the extraction-style paradigm, we separate the generated pairs or triplets from the sequence $\boldsymbol{y}'$ and ignore those invalid generations in a similar way.

We adopt the pre-trained T5 model (Raffel et al., 2020) as the generation model $f(\cdot)$, which closely follows the encoder-decoder architecture of the original Transformer (Vaswani et al., 2017). Therefore, by formulating these ABSA tasks as a text generation problem, we can tackle them in a unified sequence-to-sequence framework without task-specific model design.

	L14	R14	R15	R16
HAST+TOWE†	53.41	62.39	58.12	63.84
JERE-MHS†	52.34	66.02	59.64	67.65
SpanMlt (Zhao et al., 2020)	68.66	**75.60**	64.68	71.78
SDRN (Chen et al., 2020)	66.18	73.30	65.75	73.67
GAS-ANNOTATION-R	68.74	72.66	65.03	73.75
GAS-EXTRACTION-R	67.58	73.22	65.83	74.12
GAS-ANNOTATION	**69.55**	75.15	**67.93**	**75.42**
GAS-EXTRACTION	68.08	74.12	67.19	74.54

Table 1: Main results of the AOPE task. The best results are in bold, second best results are underlined. Results are the average F1 scores over 5 runs. † denotes results are from Zhao et al. (2020).

	L14	R14	R15	R16
BERT+GRU (Li et al., 2019b)	61.12	73.17	59.60	70.21
SPAN-BERT (Hu et al., 2019)	61.25	73.68	62.29	-
IMN-BERT (He et al., 2019)	61.73	70.72	60.22	-
RACL (Chen and Qian, 2020)	63.40	75.42	66.05	-
Dual-MRC (Mao et al., 2021)	65.94	75.95	65.08	-
GAS-ANNOTATION-R	67.37	75.77	65.75	71.87
GAS-EXTRACTION-R	66.71	76.30	64.00	72.39
GAS-ANNOTATION	**68.64**	76.58	**66.78**	73.21
GAS-EXTRACTION	68.06	**77.13**	65.96	**73.64**

Table 2: Main results of the UABSA task. The best results are in bold, second best results are underlined. Results are the average F1 scores over 5 runs.

2.3 Prediction Normalization

Ideally, the generated element $e \in \boldsymbol{s}$ after decoding is supposed to exactly belong to the vocabulary set it is meant to be. For example, the predicted aspect term should explicitly appear in the input sentence. However, this might not always hold since each element is generated from the vocabulary set containing all tokens instead of its specific vocabulary set. Thus, the predictions of a generation model may exhibit morphology shift from the ground-truths, *e.g.*, from *single* to *plural* nouns.

	L14	R14	R15	R16
CMLA+ (Wang et al., 2017)	33.16	42.79	37.01	41.72
Li-unified-R (Li et al., 2019a)	42.34	51.00	47.82	44.31
Pipeline (Peng et al., 2020)	42.87	51.46	52.32	54.21
Jet (Xu et al., 2020)	43.34	58.14	52.50	63.21
Jet+BERT (Xu et al., 2020)	51.04	62.40	57.53	63.83
GAS-ANNOTATION-R	52.80	67.35	56.95	67.43
GAS-EXTRACTION-R	58.19	70.52	60.23	69.05
GAS-ANNOTATION	54.31	69.30	61.02	68.65
GAS-EXTRACTION	**60.78**	**72.16**	**62.10**	**70.10**

Table 3: Main results of the ASTE task. The best results are in bold, second best results are underlined. Results are the average F1 scores over 5 runs.

	Rest15	Rest16
Baseline (Brun and Nikoulina, 2018)	-	38.10
TAS-LPM-CRF (Wan et al., 2020)	54.76	64.66
TAS-SW-CRF (Wan et al., 2020)	57.51	65.89
TAS-SW-TO (Wan et al., 2020)	58.09	65.44
GAS-ANNOTATION-R	59.27	66.54
GAS-EXTRACTION-R	60.63	68.31
GAS-ANNOTATION	60.06	67.70
GAS-EXTRACTION	**61.47**	**69.42**

Table 4: Main results of the TASD task. The best results are in bold, second best results are underlined. Results are the average F1 scores over 5 runs.

We propose a prediction normalization strategy to refine the incorrect predictions resulting from such issue. For each sentiment type c denoting the type of the element e such as the aspect term or sentiment polarity, we first construct its corresponding vocabulary set V_c. For aspect term and opinion term, V_c contains all words in the current input sentence $\boldsymbol{x}$; for aspect category, V_c is a collection of all categories in the dataset; for sentiment polarity, V_c contains all possible polarities. Then for a predicted element e of the sentiment type c, if it does not belong to the corresponding vocabulary set V_c, we use $\bar{e} \in V_c$, which has the smallest Levenshtein distance (Levenshtein, 1966) with e, to replace e.

3 Experiments

3.1 Experimental Setup

Datasets We evaluate the proposed GAS framework on four popular benchmark datasets including `Laptop14`, `Rest14`, `Rest15`, and `Rest16`, originally provided by the SemEval shared challenges (Pontiki et al., 2014, 2015, 2016). For each ABSA task, we use the public datasets derived from them with more sentiment annotations. Specifically, we adopt the dataset provided by Fan et al. (2019), Li et al. (2019a), Xu et al. (2020), Wan et al. (2020) for the AOPE, UABSA, ASTE, TASD task respectively. For a fair comparison, we use the same data split as previous works.

Evaluation Metrics We adopt F1 scores as the main evaluation metrics for all tasks. A prediction is correct if and only if all its predicted sentiment elements in the pair or triplet are correct.

Experiment Details We adopt the T5 base model from *huggingface* Transformer library[2] for all experiments. T5 closely follows the original encoder-decoder architecture of the Transformer model, with some slight differences such as different position embedding schemes. Therefore, the encoder and decoder of it have similar parameter size as the BERT-BASE model. For all tasks, we use similar experimental settings for simplicity: we train the model with the batch size of 16 and accumulate gradients every two batches. The learning rate is set to be 3e-4. The model is trained up to 20 epochs for the AOPE, UABSA, and ASTE task and 30 epochs for the TASD task.

[2] https://github.com/huggingface/transformers

3.2 Main Results

The main results for the AOPE, UABSA, ASTE, TASD task are reported in Tables 1, 2, 3, 4 respectively. For our proposed GAS framework, we also present the raw results without the proposed prediction normalization strategy (with the suffix "-R"). All results are the average F1 scores across 5 runs with different random seeds.

It is noticeable that our proposed methods, based on either annotation-style or extraction-style modeling, establish new state-of-the-art results in almost all cases. The only exception is on the `Rest15` dataset for the AOPE task, our method is still on par with the previous best performance. It shows that tackling various ABSA tasks with the proposed unified generative method is an effective solution. Moreover, we can see that our method performs especially well on the ASTE and TASD tasks, the proposed extraction-style method outperforms the previous best models by 7.6 and 3.7 average F1 scores (across different datasets) on them respectively. It implies that incorporating the label semantics and appropriately modeling the interactions among those sentiment elements are essential for tackling complex ABSA problems.

	BEFORE	AFTER	LABEL
#1	*Bbq rib*	*BBQ rib*	*BBQ rib*
#2	*repeat*	*repeats*	*repeats*
#3	*chicken peas*	*chick peas*	*chick peas*
#4	*bodys*	*bodies*	None
#5	*cafe*	*coffee*	*coffee*
#6	*vegetarian*	*vegan*	*vegetarian*
#7	*salmon*	*not*	*spinach*
#8	*flight cookie*	*might cookie*	*fortune cookie*

Table 5: Example cases of the predictions before and after the prediction normalization.

3.3 Discussions

Annotation-style & Extraction-style As shown in result tables, the annotation-style method generally performs better than the extraction-style method on the AOPE and UASA task. However, the former one becomes inferior to the latter on the more complex ASTE and TASD tasks. One possible reason is that, on the ASTE and TASD tasks, the annotation-style method introduces too much content, such as the aspect category and sentiment polarity, into the target sentence, which increases the difficulty of sequence-to-sequence learning.

Why Prediction Normalization Works To better understand the effectiveness of the proposed prediction normalization strategy, we randomly sample some instances from the ASTE task that have different raw prediction and normalized prediction (*i.e.*, corrected by our strategy). The predicted sentiment elements before and after the normalization, as well as the gold label of some example cases are shown in Table 5. We find that the normalization mainly helps on two occasions: The first one is the morphology shift where two words have minor lexical differences. For example, the method fixes "*Bbq rib*" to "*BBQ rib*" (#1) and "*repeat*" to "*repeats*" (#2). Another case is orthographic alternatives where the model might generate words with the same etyma but different word types, *e.g.*, it outputs "*vegetarian*" rather than "*vegan*" (#6). Our proposed prediction normalization, which finds the replacement from the corresponding vocabulary set via Levenshtein distance, is a simple yet effective strategy to alleviate this issue.

We also observe that our prediction strategy may fail if the raw predictions are quite lexically different or even semantically different from the gold-standard labels (see Case #4, #7 and #8). In these cases, the difficulty does not come from the way of performing prediction normalization but the generation of labels close to the ground truths, especially for the examples containing implicit aspects or opinions (Case #4).

4 Conclusions and Future Work

We tackle various ABSA tasks in a novel generative framework in this paper. By formulating the target sentences with our proposed annotation-style and extraction-style paradigms, we solve multiple sentiment pair or triplet extraction tasks with a unified generation model. Extensive experiments on multiple benchmarks across four ABSA tasks show the effectiveness of our proposed method.

Our work is an initial attempt on transforming ABSA tasks, which are typically treated as classification problems, into text generation problems. Experimental results indicate that such transformation is an effective solution to tackle various ABSA tasks. Following this direction, designing more effective generation paradigms and extending such ideas to other tasks can be interesting research problems for future work.

References

Ben Athiwaratkun, Cícero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. Augmented natural language for generative sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 375–385.

Caroline Brun and Vassilina Nikoulina. 2018. Aspect based sentiment analysis into the wild. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2018*, pages 116–122.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 452–461.

Shaowei Chen, Jie Liu, Yu Wang, Wenzheng Zhang, and Ziming Chi. 2020. Synchronous double-channel recurrent network for aspect-opinion pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 6515–6524.

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, ACL 2020, pages 3685–3694.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 2509–2518.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 504–515.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 537–546.

Binxuan Huang and Kathleen M. Carley. 2018. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 6279–6284.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 6714–6721.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4194–4200.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019b. Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019*, pages 34–41.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1433–1443.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: dual cross-shared RNN for aspect term-polarity co-extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 591–601.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 3538–3547.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. *CoRR*, abs/2101.00816.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8600–8607.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval@NAACL-HLT*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval@COLING 2014*, pages 27–35.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

Information Processing Systems 2017, pages 5998–6008.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 9122–9129.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3316–3322.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 606–615.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 2339–2349.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, pages 2979–2985.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 3540–3549.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 3239–3248.

Bilingual Mutual Information Based Adaptive Training for Neural Machine Translation

Yangyifan Xu[1]*, Yijin Liu[1,2], Fandong Meng[2], Jiajun Zhang[3,4], Jinan Xu[1]† and Jie Zhou[2]
[1]Beijing Jiaotong University, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[3]National Laboratory of Pattern Recognition, Institute of Automation, CAS
[4]School of Artificial Intelligence, University of Chinese Academy of Sciences
xuyangyifan2021@ia.ac.cn, adaxry@gmail.com
{fandongmeng, withtomzhou}@tencent.com
jjzhang@nlpr.ia.ac.cn, jaxu@bjtu.edu.cn

Abstract

Recently, token-level adaptive training has achieved promising improvement in machine translation, where the cross-entropy loss function is adjusted by assigning different training weights to different tokens, in order to alleviate the token imbalance problem. However, previous approaches only use static word frequency information in the target language without considering the source language, which is insufficient for bilingual tasks like machine translation. In this paper, we propose a novel bilingual mutual information (BMI) based adaptive objective, which measures the learning difficulty for each target token from the perspective of bilingualism, and assigns an adaptive weight accordingly to improve token-level adaptive training. This method assigns larger training weights to tokens with higher BMI, so that easy tokens are updated with coarse granularity while difficult tokens are updated with fine granularity. Experimental results on WMT14 English-to-German and WMT19 Chinese-to-English demonstrate the superiority of our approach compared with the Transformer baseline and previous token-level adaptive training approaches. Further analyses confirm that our method can improve the lexical diversity.

1 Introduction

Neural machine translation (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017; Chen et al., 2018; Meng and Zhang, 2019; Zhang et al., 2019; Yan et al., 2020; Liu et al., 2021) has achieved remarkable success. As a data-driven model, the performance of NMT depends on training corpus. Balanced training data is a crucial factor in building a superior model. However, natural languages conform to the Zipf's law (Zipf, 1949), the frequencies of words exhibit the long tail characteristics, which brings an imbalance in the distribution of words in training corpora. Some studies (Jiang et al., 2019; Gu et al., 2020) assign different training weights to target tokens according to their frequencies. These approaches alleviate the token imbalance problem and indicate that tokens should be treated differently during training.

However, there are two issues in existing approaches. First, these approaches believe that low-frequency words are not sufficiently trained and thus amplify the weight of them. Nevertheless, low-frequency tokens are not always difficult as the model competence increases (Wan et al., 2020). Second, previous studies only use monolingual word frequency information in the target language without considering the source language, which is insufficient for bilingual tasks, e.g., machine translation. The mapping between bilingualism is a more appropriate indicator. As shown in Table 1, word frequency of *pleasing* and *bearings* are both 847. Corresponding to Chinese, *pleasing* has multiple mappings, while *bearings* is relatively single. The more multivariate the mapping is, the less confidence in predicting the target word given the source context. He et al. (2019) also confirm this view that words with multiple mappings contribute more to the BLEU score.

To tackle the above issues, we propose bilingual mutual information (BMI), which has two characteristics: 1) BMI measures the learning difficulty for each target token by considering the strength of association between it and the source sentence; 2) for each target token, BMI can dynamically adjust according to the context. BMI-based adaptive training can dynamically adjust the learning granularity on tokens. Easy tokens are updated with coarse granularity while difficult tokens are updated with

*This work was done when Yangyifan Xu was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China
†Jinan Xu is the corresponding author of the paper.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 511–516
August 1–6, 2021. ©2021 Association for Computational Linguistics

pleasing (847)	gāoxìng (81); yúkuài (74); xǐyuè (63); qǔyuè (49) ...
bearings (847)	zhóuchéng (671) ...

Table 1: An example from the WMT19 Chinese-English training set. The Chinese words are presented in pinyin style and the word frequency is shown in brackets. The two words have the same word frequency, while the mapping of *bearings* is more stable than that of *pleasing*.

fine granularity.

We evaluate our approach on both WMT14 English-to-German and WMT19 Chinese-to-English translation tasks. Experimental results on two benchmarks demonstrate the superiority of our approach compared with the Transformer baseline and previous token-level adaptive training approaches. Further analyses confirm that our method can improve the lexical diversity. The main contributions[1] of this paper can be summarized as follows:

- We propose a training objective based on bilingual mutual information (BMI), which can reflects the learning difficulty for each target token from the perspective of bilingualism, and assigns an adaptive weight accordingly to guide the adaptive training of machine translation.
- Experimental results show that our method can improve not only the machine translation quality, but also the lexical diversity.

2 Background

2.1 Neural Machine Translation

A NMT system is a neural network that translates a source sentence $\mathbf{x}$ with n words to a target sentence $\mathbf{y}$ with m words. During the training process, NMT models are optimized by minimizing cross entropy:

$$\mathcal{L} = -\frac{1}{m}\sum_{j=1}^{m} \log p(y_j|\mathbf{y}_{<j}, \mathbf{x}), \quad (1)$$

where y_j is the ground-truth token at the j-th position and $\mathbf{y}_{<j}$ is the translation history known before predicting token y_j.

[1]Reproducible code: https://github.com/xydaytoy/BMI-NMT

BMI= 2.29	In ball **bearings**, as the radial clearance increases , the axial clearance increases as well. zài qiú **zhóu chéng**, jìng xiàng jiàn xì de zēng jiā, zhóu xiàng yóu xì zēng zhǎng.
BMI= 1.83	One of his crowd **pleasing** notions is that migrants will infect Americans with terrible diseases. tā **qǔ yuè** qún méng de gài niàn zhī yī shì yí mín huì jǐ měi guó rén dài lái kě pà de chuán rǎn bìng.

Figure 1: An example from WMT19 Chinese-to-English training set. Words with Red and Bold fonts have the same word frequency while different BMI.

2.2 Token-level Adaptive Training Objective

Following (Gu et al., 2020), the token-level adaptive training objective is

$$\mathcal{L} = -\frac{1}{m}\sum_{j=1}^{m} w_j \cdot \log p(y_j|\mathbf{y}_{<j}, \mathbf{x}), \quad (2)$$

where w_j is the weight assigned to the target token y_j . Gu et al. (2020) used monolingual word frequency information in the target language to calculate the w_j. The weight does not contain the information of the source language, and cannot be dynamically adjusted with the context.

3 BMI-based Adaptive Training

In this section, we start with the definition of the bilingual mutual information (BMI). Then we analyze the relationship between BMI and translation difficulty. Based on this, we introduce our BMI-based token-level adaptive training objective.

3.1 Definition of BMI

Mutual information measures the strength of association between two random variables by comparing the number of their individual and joint occurrences. We develop BMI, which is calculated by summarizing the mutual information of the target token and each token in the source sentence, to measure the learning difficulty of the model. Token pairs with high BMI are considered easy, since they have high co-occurrence relative to the frequency. Given the source sentence $\mathbf{x}$ and target token y_j, we define the bilingual mutual information as[2]:

$$\mathrm{BMI}(\mathbf{x}, y_j) = \sum_{i=1}^{n} \log \frac{f(x_i, y_j)}{f(x_i) \cdot f(y_j)/K}, \quad (3)$$

[2]To ensure comparability of two probability distribution, the tokens that appear multiple times in a sentence and the token pairs that appear multiple times in a sentence pair are not counted repeatedly.

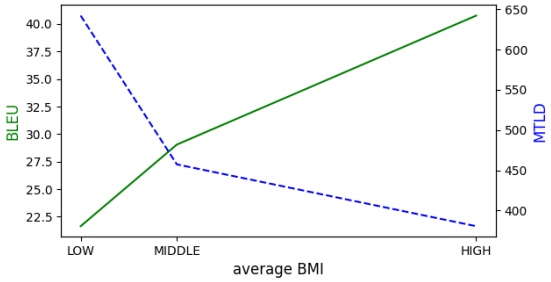


Figure 2: The BLEU (green solid line) and MTLD (blue dotted line) values on the subsets of WMT14 English-German training set, divided according to the average BMI. All target sentences of the training set are divided into three subsets according to the average BMI of the tokens in the sentence, which are equal in size and denoted as LOW, MIDDLE, and HIGH, respectively. BLEU indicates the learning difficulty of the model. The measure of textual lexical diversity (MTLD) (McCarthy and Jarvis, 2010) represents the lexical diversity of the data set. The results show that high BMI means relatively stable mapping, which is easy to be learned by the model and has low lexical diversity.

where $f(x_i)$ and $f(y_j)$ are total number of sentences in the corpus containing at least one occurrence of x_i and y_j, respectively, $f(x_i, y_j)$ represents total number of sentences in the corpus having at least one occurrence of the word pair (x_i, y_j), and K denotes total number of sentences in the corpus.

3.2 What BMI Measures?

We use an example to illustrate our idea. Figure 1 shows two sentence pairs. Words with Red and Bold fonts have the same word frequency. As shown in Table 1, *pleasing* has multiple mappings, while the mapping of *bearings* is relatively single. As a result, the appearance of corresponding English word brings different confidence of the appearance of the Chinese word, which can be reflected by BMI. Further statistical results are shown in Figure 2, high BMI means relatively stable mapping, which is easy to be learned by the model and has low lexical diversity.

3.3 BMI-based Objective

We calculate the token-level weight by scaling BMI and adjusting the lower limit as follows:

$$w_j = S \cdot \mathrm{BMI}(\mathbf{x}, y_j) + B. \quad (4)$$

The two hyperparameters S (scale) and B (base) influence the magnitude of change and the lower limit, respectively.

In training process, the loss of simple tokens will be amplified, the model updates simple tokens with coarse granularity, because our strategy thinks the model can easily predict these target tokens given the source sentence, and it needs to increase the penalty if the prediction is wrong. For difficult tokens, the model has a higher tolerance because their translation errors may not be absolute. As a result, the loss is small due to the small weight and the difficult tokens are always updated in a fine-grained way.

4 Experiments

We evaluate our method on the Transformer (Vaswani et al., 2017) and conduct experiments on two widely-studied NMT tasks, WMT14 English-to-German (En-De) and WMT19 Chinese-to-English (Zh-En).

4.1 Data Preparation

EN-DE. The training data consists of 4.5M sentence pairs from WMT14. Each word in the corpus has been segmented into subword units using byte pair encoding (BPE) (Sennrich et al., 2016) with 32k merge operations. The vocabulary is shared among source and target languages. We select newstest2013 for validation and report the BLEU scores on newstest2014.

ZH-EN. The training data is from WMT19 which consists of 20.5M sentence pairs. The number of merge operations in byte pair encoding (BPE) is set to 32K for both source and target languages. We use newstest2018 as our validation set and newstest2019 as our test set, which contain 4k and 2k sentences, respectively.

4.2 Systems

Transformer. We implement our approach with the open source toolkit THUMT (Zhang et al., 2017) and strictly follow the setting of Transformer-Base in (Vaswani et al., 2017).

Exponential (Gu et al., 2020). This method adds an additional training weights to low-frequency target tokens:

$$w_j = A \cdot e^{-T \cdot Count(y_j)} + 1. \quad (5)$$

Chi-Square (Gu et al., 2020). The weighting function of this method is similar to the form of chi-square distribution

$$w_j = A \cdot Count^2(y_j) e^{-T \cdot Count(y_j)} + 1. \quad (6)$$

	B	S	EN-DE	ZH-EN
BMI	1.0	0.05	26.87	23.52
		0.10	26.89	**23.61**
		0.15	26.93	23.49
		0.20	26.98	23.39
		0.25	26.91	23.24
		0.30	26.85	23.50
	0.9	0.15	26.93	23.31
		0.20	26.88	23.31
		0.25	26.96	23.41
	0.8	0.15	**27.01**	23.40
		0.20	26.81	23.25
		0.25	26.93	23.50
	0.7	0.15	26.92	23.44
		0.20	26.90	23.35
		0.25	26.89	23.34

Table 2: Performance of our methods on the validation sets with different hyperparameters S and B.

BMI. Our system is first trained with normal cross entropy loss (Equation 1) for 100k steps. Then the model is further trained with BMI-based adaptive objective (Equation 4) for 100k steps. The same procedure was used for the competing methods. In order to eliminate the influence of noise, we assign the weight of tokens with BMI lower than 0.4 to zero during the training process.

4.3 Hyperparameters

We introduce two hyperparameters, B and S, to adjust the weight distribution based on BMI, as shown in Equation 4. In our experiments, we fixed B to narrow search space [0.7, 1]. We tuned another hyperparameter S on the validation sets. The results are shown in Table 2. Finally, we use the best hyperparameters found on the validation set for the final evaluation of the test set. For En-De, B = 0.8 and S = 0.15, for Zh-En, B = 1.0 and S = 0.1.

4.4 Main Results

As shown in Table 3, compared with (Vaswani et al., 2017), our Transformer outperforms it by 0.67 BLEU points. We use a strong baseline system in this work in order to make the evaluation convincing. Improvement of existing methods (Gu et al., 2020) is limited over strong baseline. Exponential objective achieves 28.17 (+0.2) BLEU on En-De and Chi-Square objective achieves 24.62 (+0.25) BLEU on Zh-En. Our method yields 28.53 (+0.56) and 25.19 (+0.82) BLEU on the En-De task and Zh-En task, respectively. The significant and consistent improvement on the two large-scale dataset demonstrates the effectiveness of our method.

System	EN-DE	ZH-EN
Existing NMT systems		
Vaswani et al. (2017)	27.3	-
Chi-Square	27.51	-
Exponential	27.60	-
Our NMT systems		
Transformer	27.97	24.37
+ Chi-Square	28.08(+*0.11*)	24.62(+*0.25*)
+ Exponential	28.17(+*0.20*)	24.33(-*0.04*)
+ BMI	**28.53(+*0.56*)***	**25.19(+*0.82*)***

Table 3: BLEU scores (%) on the WMT14 En-De test set and the WMT19 Zh-En test set. Results of our method marked with '*' are statistically significant (Koehn, 2004) by contrast to all other models ($p<0.01$).

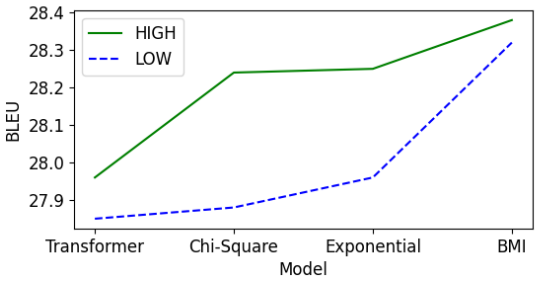


Figure 3: BLEU scores (%) on different WMT14 En-De test subsets which are grouped by their average BMI. Sentences in the HIGH subset contains more tokens with high BMI.

4.5 Results on Different BMI Intervals

We score each target sentence of newstest2014 by calculating the average BMI of each token in the sentence, and then divide newstest2014 into two subsets with equal size according to the score, denoted as HIGH and LOW, respectively. As shown in Figure 3, compared to Transformer, frequency-based methods outperform on the HIGH subset but have no obvious improvement on the LOW subset. By contrast, our method can not only bring a stable improvement on the HIGH subset, the improvement is even more obvious on the LOW subset. Low BMI means relatively rich mapping. We believe that the model should have a higher tolerance for these tokens because their translation errors may not be absolute. For example, the model outputs another token with similar meaning. Therefore, our method improves more on LOW subset.

Models	MATTR	HD-D	MTLD
Transformer	89.41	94.05	230.36
+ Chi-Square	89.37	94.02	230.02
+ Exponential	89.41	94.08	232.98
+ BMI	**89.45**	**94.10**	**236.43**
Reference	90.92	94.88	259.98

Table 4: The lexical diversity of WMT14 En-De translations measured by MATTR (%), HD-D (%) and MTLD. A larger value means a higher diversity.

4.6 Effects on Lexical Diversity

Vanmassenhove et al. (2019) suggest that the vanilla NMT systems exacerbate bias presented in corpus, resulting in lower vocabulary diversity. We use three measures of lexical diversity, namely, moving-average type-token ratio (MATTR) (Covington and McFall, 2010), the approximation of hypergeometric distribution (HD-D) and the measure of textual lexical diversity (MTLD) (McCarthy and Jarvis, 2010). Results in Table 4 show that, on improving the lexical diversity of translation, our method is superior to existing methods (Chi-Square and Exponential) based on word frequency.

4.7 Contrast with Label Smoothing

There are similarities between token-level adaptive training and label smoothing, because they both adjust the loss function of the model by token weighting. In particular, for some smoothing methods guided by prior or posterior knowledge of training data (Gao et al., 2020; Pereyra et al., 2017), different tokens are treated differently. But these similarities are not the key points of the two methods, and they are essentially different. The first and very important point is that the motivations of the two methods are different. Label smoothing is a regularization method to avoid overfitting, while our method treats samples of different difficulty differently for adaptive training. Second, the two methods work in different ways. Label smoothing is used when calculating the cross-entropy loss. It emphasizes how to assign the weight of tokens other than the golden one, and indirectly affects the training of the golden token. While our method is used after calculating the cross-entropy loss. It is calculated according to the golden token at each position in the reference, which is more direct. In all experiments, we employed uniform label smoothing of value $\epsilon_{ls} = 0.1$, the results show that the two methods does not conflict when used together.

5 Conclusion

We propose a novel bilingual mutual information based adaptive training objective, which can measure the learning difficulty for each target token from the perspective of bilingualism, and adjust the learning granularity dynamically to improve token-level adaptive training. Experimental results on two translation tasks show that our method can bring a significant improvement in translation quality, especially on sentences that are difficult to learn by the model. Further analyses confirm that our method can also improve the lexical diversity.

Acknowledgments

The research work descried in this paper has been supported by the National Key R&D Program of China (2020AAA0108001), the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130), the Beijing Academy of Artificial Intelligence (BAAI2019QN0504) and the Youth Innovation Promotion Association CAS No. 2017172. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Michael A Covington and Joe D McFall. 2010. Cutting the gordian knot: The moving-average type–token ratio (mattr). *Journal of quantitative linguistics*, 17(2):94–100.

Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. 2020. Towards a better understanding of label smoothing in neural machine translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 212–223.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. 2020. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1035–1046.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards understanding neural machine translation with word importance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 952–961.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. In *The World Wide Web Conference*, pages 2879–2885.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Yijin Liu, Fandong Meng, Jie Zhou, Yufeng Chen, and Jinan Xu. 2021. Faster depth-adaptive transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Philip M McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Fandong Meng and Jinchao Zhang. 2019. DTMT: A novel deep transition architecture for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 224–231.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, pages 3104–3112. Curran Associates, Inc.

Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. 2019. Lost in translation: Loss and decay of linguistic richness in machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 222–232, Dublin, Ireland. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Yu Wan, Baosong Yang, Derek F Wong, Yikai Zhou, Lidia S Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080.

Jianhao Yan, Fandong Meng, and Jie Zhou. 2020. Multi-unit transformers for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1047–1059, Online.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, and Yang Liu. 2017. Thumt: An open source toolkit for neural machine translation. *arXiv preprint arXiv:1706.06415*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343.

George Kingsley Zipf. 1949. Human behavior and the principle of least effort.

Continual Learning for Task-oriented Dialogue System with Iterative Network Pruning, Expanding and Masking

Binzong Geng[1,2]*, **Fajie Yuan**[3], **Qiancheng Xu**[4], **Ying Shen**[5], **Ruifeng Xu**[6], **Min Yang**[2]†
[1]University of Science and Technology of China
[2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[3]Westlake University [4]Georgia Institute of Technology
[5]Sun Yat-sen University [6]Harbin Institute of Technology (Shenzhen)
{bz.geng,min.yang}@siat.ac.cn, yuanfajie@westlake.edu.cn
qxu309@gatech.edu, sheny76@mail.sysu.edu.cn, xuruifeng@hit.edu.cn

Abstract

This ability to learn consecutive tasks without forgetting how to perform previously trained problems is essential for developing an online dialogue system. This paper proposes an effective continual learning for the task-oriented dialogue system with iterative network pruning, expanding and masking (TPEM), which preserves performance on previously encountered tasks while accelerating learning progress on subsequent tasks. Specifically, TPEM (i) leverages network pruning to keep the knowledge for old tasks, (ii) adopts network expanding to create free weights for new tasks, and (iii) introduces task-specific network masking to alleviate the negative impact of fixed weights of old tasks on new tasks. We conduct extensive experiments on seven different tasks from three benchmark datasets and show empirically that TPEM leads to significantly improved results over the strong competitors. For reproducibility, we submit the code and data at: https://github.com/siat-nlp/TPEM.

1 Introduction

Building a human-like task-oriented dialogue system is a long-term goal of AI. Great endeavors have been made in designing end-to-end task-oriented dialogue systems (TDSs) with sequence-to-sequence (Seq2Seq) models (Eric and Manning, 2017; Madotto et al., 2018; Gangi Reddy et al., 2019; Qin et al., 2020; Mi et al., 2019; He et al., 2020; Wang et al., 2020; Qin et al., 2021), which have taken the state-of-the-art of TDSs to a new level. Generally, Seq2Seq models leverage an encoder to create a vector representation of dialogue history and KB information, and then pass this representation into a decoder so as to output a response word by word. For example, GLMP (Wu et al., 2019) is a representative end-to-end TDS, which incorporates KB information into Seq2Seq model by using a global memory pointer to filter irrelevant KB knowledge and a local memory pointer to instantiate entity slots.

Despite the remarkable progress of previous works, the current dominant paradigm for TDS is to learn a Seq2Seq model on a given dataset specifically for a particular purpose, which is referred to as isolated learning. Such learning paradigm is theoretically of limited success in accumulating the knowledge it has learned before. When a stream of domains or functionalities are joined to be trained sequentially, isolated learning faces catastrophic forgetting (McCloskey and Cohen, 1989; Yuan et al., 2020, 2021). In contrast, humans retain and accumulate knowledge throughout their lives so that they become more efficient and versatile facing new tasks in future learning (Thrun, 1998). If one desires to create a human-like dialogue system, imitating such a lifelong learning skill is quite necessary.

This paper is motivated by the fact that a cognitive AI has continual learning ability by nature to develop a task-oriented dialogue agent that can accumulate knowledge learned in the past and use it seamlessly in new domains or functionalities. Continual learning (Parisi et al., 2019; Wu et al., 2018; Yuan et al., 2020, 2021) is hardly a new idea for machine learning, but remains as a non-trivial step for building empirically successful AI systems. It is essentially the case for creating a high-quality TDS. On the one hand, a dialogue system is expected to reuse previously acquired knowledge, but focusing too much on stability may hinder a TDS from quickly adapting to a new task. On the other hand, when a TDS pays too much attention to plasticity, it may quickly forget previously-acquired abilities (Mallya and Lazebnik, 2018).

In this paper, we propose a continual learning

*This work was conducted when Binzong Geng was an intern at SIAT, Chinese Academy of Sciences.
†Min Yang is corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 517–523
August 1–6, 2021. ©2021 Association for Computational Linguistics

method for task-oriented dialogue system with iterative network pruning, expanding and masking (TPEM), which preserves performance on previously encountered tasks while accelerating learning progress on the future tasks. Concretely, TPEM adopts the global-to-local memory pointer networks (GLMP) (Wu et al., 2019) as the base model due to its powerful performance in literature and easiness for implementation. We leverage iterative pruning to keep old tasks weights and thereby avoid forgetting. Meanwhile, a network expanding strategy is devised to gradually create free weights for new tasks. Finally, we introduce a task-specific binary matrix to mask some old task weights that may hinder the learning of new tasks. It is noteworthy that TPEM is model-agnostic since the pruning, expanding and binary masking mechanisms merely work on weight parameters (weight matrices) of GLMP.

We conduct extensive experiments on seven different domains from three benchmark TDS datasets. Experimental results demonstrate that our TPEM method significantly outperforms strong baselines for task-oriented dialogue generation in continual learning scenario.

2 Our Methodology

2.1 Task Definition

Given the dialogue history X and KB tuples B, TDS aims to generate the next system response Y word by word. Suppose a lifelong TDS model that can handle domains 1 to k has been built, denoted as $\mathcal{M}_{1:k}$. The goal of TDS in continual learning scenario is to train a model $\mathcal{M}_{1:k+1}$ that can generate responses of the $k+1$-th domain without forgetting how to generate responses of previous k domains. We use the terms "domain" and "task" interchangeably, because each of our tasks is from a different dialogue domain.

2.2 Overview

In this paper, we adopt the global-to-local memory pointer networks (GLMP) (Wu et al., 2019) as base model, which has shown powerful performance in TDS. We propose a continual learning method for TDS with iterative pruning, expanding, and masking. In particular, we leverage pruning to keep the knowledge for old tasks. Then, we adopt network expanding to create free weights for new tasks. Finally, a task-specific binary mask is adopted to mask part of old task weights, which may hinder the learning of new tasks. The proposed model is model-agnostic since the pruning, expanding and binary masking mechanisms merely work on weight parameters (weight matrices) of the encoder-decoder framework. Next, we will introduce each component of our TPEM framework in detail.

2.3 Preliminary: The GLMP Model

GLMP contains three primary components: external knowledge, a global memory encoder, and a local memory decoder. Next, we will briefly introduce the three components of GLMP. The readers can refer to (Wu et al., 2019) for the implementation details.

External Knowledge To integrate external knowledge into the Seq2Seq model, GLMP adopts the end-to-end memory networks to encode the word-level information for both dialogue history (dialogue memory) and structural knowledge base (KB memory). Bag-of-word representations are utilized as the memory embeddings for two memory modules. Each object word is copied directly when a memory position is pointed to.

Global Memory Encoder We convert each input token of dialogue history into a fixed-size vector via an embedding layer. The embedding vectors go through a bi-directional recurrent unit (BiGRU) (Chung et al., 2014) to learn contextualized dialogue representations. The original memory representations and the corresponding implicit representations will be summed up, so that these contextualized representations can be written into the dialogue memory. Meanwhile, the last hidden state of dialogue representations is used to generate two outputs (i.e., global memory pointer and memory readout) by reading out from the external knowledge. Note that an auxiliary multi-label classification task is added to train the global memory pointer as a multi-label classification task.

Local Memory Decoder Taking the global memory pointer, encoded dialogue history and KB knowledge as input, a sketch GRU is applied to generate a sketch response Y^s that includes the sketch tags rather than slot values. If a sketch tag is generated, the global memory pointer is then passed to the external knowledge and the retrieved object word will be picked up by the local memory pointer; otherwise, the output word is generated by the sketch GRU directly.

To effectively transfer knowledge for subsequent tasks and reduce the space consumption, the global memory encoder and external knowledge in GLMP are shared among all tasks, while a separate local memory decoder is learned by each task.

2.4 Continual Learning for TDS

We employ an iterative network pruning, expanding and masking framework for TDS in continual learning scenario, inspired by (Mallya and Lazebnik, 2018; Mallya et al., 2018).

Network Pruning To avoid "catastrophic forgetting" of GLMP, a feasible way is to retain the acquired old-task weights and enlarge the network by adding weights for learning new tasks. However, as the number of tasks grows, the complexity of model architecture increases rapidly, making the deep model difficult to train. To avoid constructing a huge network, we compress the model for the current task by releasing a certain fraction of neglectable weights of old tasks (Frankle and Carbin, 2019; Geng et al., 2021).

Suppose that for task k, a compact model $\mathcal{M}_{1:k}$ that is able to deal with tasks 1 to k has been created and available. We then free up a certain fraction of neglectable weights (denoted as $\mathbf{W}_k^F$) that have the lowest absolute weight values by setting them to zero. The released weights associated with task k are extra weights which can be utilized repeatedly for learning newly coming tasks. However, pruning a network suddenly changes the network connectivity and thereby leads to performance deterioration. To regain its original performance after pruning, we re-train the preserved weights for a small number of epochs. After a period of pruning and re-training, we obtain a sparse network with minimal performance loss on the performance of task k. This network pruning and re-training procedures are performed iteratively for learning multiple subsequent tasks. When inferring task k, the released weights are masked in a binary on/off fashion such that the network state keeps consistent with the one learned during training.

Network Expanding The amount of preserved weights for old tasks becomes larger with the growth of new tasks, and there will be fewer free weights for learning new tasks, resulting in slowing down the learning process and making the found solution non-optimal. An intuitive solution is to expand the model while learning new tasks so as to increase new capacity of the GLMP model for subsequent tasks (Hung et al., 2019b,a).

To effectively perform network expansion while keeping the compactness of network architecture, we should consider two key factors: (1) the proportion of free weights for new tasks (denoted as F_k) and (2) the number of training batches (denoted as N_k). Intuitively, it is difficult to optimize the parameters that are newly added and randomly initialized with a small number of training data. To this end, we define the following strategy to expand the hidden size H_k for the k-th task from H_{k-1}:

$$H_k = H_{k-1} + \alpha * (P_{k-1} - F_k) * \log(1 + N_k/\beta) \quad (1)$$

where α and β are two hyperparameters. P_{k-1} is the pruning ratio of task $k-1$. In this way, we are prone to expand more weights for the tasks that have less free weights but more training data.

Network Masking The preserved weights $\mathbf{W}_k^P$ of old tasks are fixed so as to retain the performance of learned tasks and avoid forgetting. However, not all preserved weights are beneficial to learn new tasks, especially when there is a large gap between old and new tasks. To resolve this issue, we apply a learnable binary mask $\mathbf{M}^k$ for each task k to filter some old weights that may hinder the learning of new tasks. We additionally maintain a matrix $\tilde{\mathbf{M}}^k$ of real-valued mask weights, which has the same size as the weight matrix $\mathbf{W}$. The binary mask matrix $\mathbf{M}^k$, which participates in forward computing, is obtained by passing each element of $\tilde{\mathbf{M}}^k$ through a binary thresholding function:

$$\mathbf{M}_{ij}^k = \begin{cases} 1, & \text{if } \tilde{\mathbf{M}}_{ij}^k > \tau \\ 0, & \text{ortherwise} \end{cases} \quad (2)$$

where τ is a pre-defined threshold. The real-valued mask $\tilde{\mathbf{M}}^k$ will be updated in the backward pass via gradient descent. After obtaining the binary mask $\mathbf{M}^k$ for a given task, we discard $\tilde{\mathbf{M}}^k$ and only store $\mathbf{M}^k$. The weights selected are then represented as $\mathbf{M}^k \odot \mathbf{W}_k^P$, which get along with free weights $\mathbf{W}_k^F$ to learn new tasks. Here, $\odot$ denotes element-wise product. Note that old weights $\mathbf{W}_k^P$ are "picked" only and keep unchanged during training. Thus, old tasks can be recalled without forgetting. Since a binary mask requires only one extra bit per parameter, TPEM only introduces an approximate overhead of 1/32 of the backbone network size per parameter, given that a typical network parameter is often represented by a 32-bit float value.

Task ID	1	2	3	4	5	6	7	
Task	Schedule	Navigation	Weather	Restaurant	Hotel	Attraction	CamRest	Avg.
Ptr-Unk	0.00/23.33	0.36/14.17	1.26/12.62	1.20/21.21	1.66/16.14	0.84/19.16	8.40/39.45	1.96/20.87
Mem2Seq	0.66/23.32	3.87/23.37	3.21/38.90	1.37/14.17	0.95/10.25	0.19/4.80	10.10/43.07	2.91/22.55
GLMP	0.95/15.01	3.91/24.34	2.56/27.12	6.51/32.76	5.24/29.60	6.72/30.31	16.96/**52.85**	6.12/30.28
UCL	12.60/60.24	4.42/33.06	4.27/47.93	3.57/15.60	2.40/10.34	1.20/14.24	12.77/39.74	5.89/31.59
Re-init	16.21/64.06	9.38/42.47	11.54/50.30	8.97/**34.06**	6.52/**33.60**	3.78/18.05	16.88/48.15	10.47/41.53
Re-init-expand	15.98/64.29	9.92/40.15	11.50/54.12	**9.41**/30.98	6.07/31.54	5.80/17.56	16.60/46.42	10.75/40.72
TPEM	**16.72/67.15**	**11.95/49.74**	**13.27/55.60**	7.98/31.90	**7.07**/30.99	**9.11/33.74**	**17.60**/51.77	**11.96/45.84**
w/o Pruning	16.68/66.74	11.33/45.01	13.07/51.76	7.67/30.02	6.57/33.25	8.96/23.56	17.48/52.08	11.68/43.20
w/o Expansion	**16.72/67.15**	**11.95/49.74**	11.35/51.85	7.40/31.73	5.17/32.89	8.71/29.63	15.17/52.16	10.92/45.02
w/o Masking	**16.72/67.15**	11.35/48.48	11.88/54.25	7.29/31.79	6.21/32.59	8.42/30.78	16.71/51.35	11.23/45.20

Table 1: BLEU/Entity F1 results evaluated on the final model after all 7 tasks are visited. We use Avg. to represent the average performance of all tasks for each method.

3 Experimental Setup

Datasets Since there is no authoritative dataset for TDS in continual learning scenario, we evaluate TPEM on 7 tasks from three benchmark TDS datasets: (1) In-Car Assistant (Eric and Manning, 2017) that contains 2425/302/304 dialogues for training/validation/testing, belonging to calendar scheduling, weather query, and POI navigation domains, (2) Multi-WOZ 2.1 (Budzianowski et al., 2018) that contains 1,839/117/141 dialogues for training/validation/testing, belonging to restaurant, attraction, and hotel domains, and (3) CamRest (Wen et al., 2016) that contains 406/135/135 dialogues from the restaurant reservation domain for training/validation/testing.

Implementation Details Following (Wu et al., 2019), the word embeddings are randomly initialized from normal distribution $\mathcal{N}(0, 0.1)$ with size of 128. We set the size of encoder and decoder as 128. We conduct one-shot pruning with ratio $P = 0.5$. The hyperparameters α and β are set to 32 and 50, respectively. We use Adam optimizer to train the model, with an initial learning rate of $1e^{-3}$. The batch size is set to 32 and the number of memory hop k is set to 3. We set the maximum re-training epochs to 5. That is, we adopt the same re-training epochs for different tasks. We run our model three times and report the average results.

Baseline Methods First, we compare TPEM with three widely used TDSs: **Ptr-Unk** (Eric and Manning, 2017), **Mem2Seq** (Madotto et al., 2018), and **GLMP** (Wu et al., 2019). In addition, we also compare TPEM with **UCL** (Ahn et al., 2019) which is a popular continual learning method. Furthermore, we report results obtained by the base model when its parameters are optionally re-initialized after a task has been visited (denoted as **Re-init**). We also report the results of Re-init with network expansion (denoted as **Re-init-expand**). Different from GLMP that keeps learning a TDS by utilizing parameters learned from past tasks as initialization for the new task, both Re-init and Re-init-expand save a separate model for each task in inference without considering the continual learning scenario.

4 Experimental Results

Main Results We evaluate TPEM and baselines with BLEU (Papineni et al., 2002) and entity F1 (Madotto et al., 2018). We conduct experiments by following the common continual learning setting, where experimental data from 7 domains arrives sequentially. The results of each task are reported after all 7 tasks have been learned. That is, each model keeps learning a new task by using the weights learned from past tasks as initialization. The evaluation results are reported in Table 1. The typical TDSs (i.e., Ptr-Unk, Mem2Seq, GLMP) perform much worse than the continual learning methods (UCL and TPEM). This is consistent with our claim that conventional TDSs suffer from catastrophic forgetting. TPEM achieves significantly better results than baseline methods (including Re-init and Re-init-expand) on both new and old tasks. The improvement mainly comes from the iterative network pruning, expanding and masking.

Ablation Study To investigate the effectiveness of each component in TPEM, we conduct ablation test in terms of removing network pruning (w/o Pruning), network expansion (w/o Expansion), and network masking (w/o Masking). The experimental results are reported in Table 1. The performance

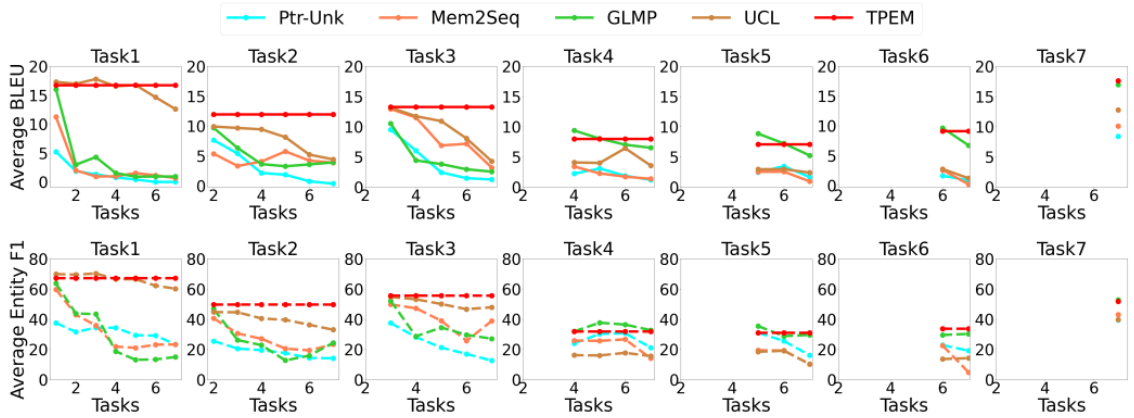


Figure 1: The change of BLEU/Entity F1 scores for each task during the whole learning process (i.e., after learning new tasks).

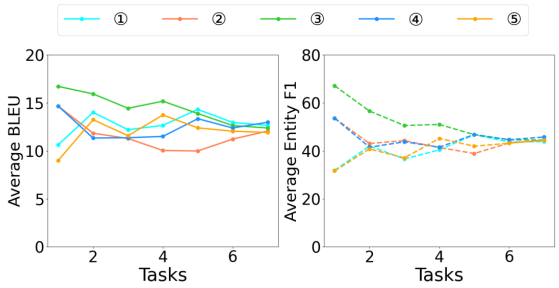


Figure 2: The average results of TPEM over 7 domains with 5 different orderings randomly sampled from the 7 domains.

of TPEM drops more sharply when discarding network pruning than discarding the other two components. This is within our expectation since the expansion and masking strategies rely on network pruning, to some extent. Not surprisingly, combining all the components achieves the best results. Furthermore, by comparing the results of Re-init and Re-init-expand, we can observe that only using network expanding cannot improve the performance of Re-init.

Case Study We provide visible analysis on the middle states of all the models. Figure 1 shows how the results of each task change as new tasks are being learned subsequently. Taking the third task as an example, we observe that the performance of conventional TDSs and UCL starts to decay sharply after learning new tasks, probably because the knowledge learned from these new tasks interferes with what was learned previously. However, TPEM achieves stable results over the whole learning process, without suffering from knowledge forgetting.

Effect of Task Ordering To explore the effect of task ordering for our TPEM model, we randomly sample 5 different task orderings in this experiment. The average results of TPEM over 7 domains with 5 different orderings are shown in Figure 2. We can observe that although our method has various behaviors with different task orderings, TPEM is in general insensitive to orders because the results show similar trends, especially for the last 2 tasks.

5 Conclusion

In this paper, we propose a continual learning method for task-oriented dialogue systems with iterative network pruning, expanding and masking. Our dialogue system preserves performance on previously encountered tasks while accelerating learning progress on subsequent tasks. Extensive experiments on 7 different tasks show that our TPEM method performs significantly better than compared methods. In the future, we plan to automatically choose the pruning ratio and the number of re-training epochs in the network pruning process for each task adaptively.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (No. 61906185), Natural Science Foundation of Guangdong Province of China (No. 2019A1515011705), Youth Innovation Promotion Association of CAS China (No. 2020357), Shenzhen Science and Technology Innovation Program (Grant No. KQTD20190929172835662), Shenzhen Basic Research Foundation (No. JCYJ20200109113441941).

References

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based continual learning with adaptive regularization. In *NeurIPS*, pages 4392–4402.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *EMNLP*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Mihail Eric and Christopher D Manning. 2017. A Copy-augmented Sequence-to-sequence Architecture Gives Good Performance on Task-oriented Dialogue. *EACL*.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Training pruned neural networks. *ICLR*.

Revanth Gangi Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. 2019. Multi-level memory for task oriented dialogs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3744–3754.

Binzong Geng, Min Yang, Fajie Yuan, Shupeng Wang, Xiang Ao, and Ruifeng Xu. 2021. Iterative network pruning with uncertainty regularization forlifelong sentiment classification. In *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*.

Wanwei He, Min Yang, Rui Yan, Chengming Li, Ying Shen, and Ruifeng Xu. 2020. Amalgamating knowledge from two teachers for task-oriented dialogue system with adversarial training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3498–3507.

Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. 2019a. Compacting, picking and growing for unforgetting continual learning. In *Advances in Neural Information Processing Systems*, pages 13647–13657.

Steven C. Y. Hung, Jia-Hong Lee, Timmy S. T. Wan, Chein-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. 2019b. Increasingly packing multiple facial-informatics modules in a unified deep-learning model via lifelong learning. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, page 339–343.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *ACL*, pages 1468–1478.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision*, pages 67–82.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *IJCAI*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Bowen Qin, Min Yang, Lidong Bing, Qingshan Jiang, Chengming Li, and Ruifeng Xu. 2021. Exploring auxiliary reasoning tasks for task-oriented dialog systems with meta cooperative learning. In *The AAAI Conference on Artificial Intelligence*.

Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *ACL*.

Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer.

Jian Wang, Junhao Liu, Wei Bi, Xiaojiang Liu, Kejing He, Ruifeng Xu, and Min Yang. 2020. Dual dynamic memory network for end-to-end multi-turn task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4100–4110.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve J. Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. *CoRR*, abs/1606.03352.

Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. 2018. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31:5962–5972.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. *CoRR*, abs/1901.04713.

Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1469–1478.

Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. 2021. One person, one model, one world: Learning continual user representation without forgetting. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

TIMERS: Document-level Temporal Relation Extraction

Puneet Mathur
University of Maryland, College Park
puneetm@umd.edu

Rajiv Jain
Adobe Research
rajijain@adobe.com

Franck Dernoncourt
Adobe Research
dernonco@adobe.com

Vlad Morariu
Adobe Research
morariu@adobe.com

Quan Hung Tran
Adobe Research
qtran@adobe.com

Dinesh Manocha
University of Maryland, College Park
dmanocha@umd.edu

Abstract

We present **TIMERS** - a **TIME**, **R**hetorical and **S**yntactic-aware model for document-level temporal relation classification. Our proposed method leverages rhetorical discourse features and temporal arguments from semantic role labels, in addition to traditional local syntactic features, trained through a Gated Relational-GCN. Extensive experiments show that the proposed model outperforms previous methods by 5-18% on the TDDiscourse, TimeBank-Dense, and MATRES datasets due to our discourse-level modeling.

1 Introduction

Temporal relation extraction (TempRel) is a challenging task that involves determining the temporal order between two events in a text (Pustejovsky et al., 2003). Understanding the temporal ordering of events in a document plays a key role in downstream tasks such as timeline creation (Leeuwenberg and Moens, 2018), time-aware summarization (Noh et al., 2020), temporal question-answering (Ning et al., 2020), and temporal information extraction (Leeuwenberg and Moens, 2019).

Prior work focuses on extracting temporal relations between event pairs (a.k.a., *TLINKS*) present in the same sentence (*Intra-sentence TLINKS*) or adjacent sentences (*Inter-sentence TLINKS*), mostly ignoring document-level pairs (*Cross-document TLINKS*) (Reimers et al., 2016). Past works have used RNN (Cheng and Miyao, 2017; Meng et al., 2017; Goyal and Durrett, 2019; Ning et al., 2019; Han et al., 2019a,c,b, 2020b) and Transformer networks (Ballesteros et al., 2020; Zhao et al., 2020b) for encoding a few sentences or a short paragraph but do not capture long-range dependencies and multi-hop reasoning at the document-level. This shortcoming is shown in the TDDiscourse dataset (Naik et al., 2019), which was designed to highlight global discourse-level challenges, e.g., multi-hop chain reasoning, future or hypothetical events, and reasoning requiring world knowledge.

We propose **TIMERS** - a **TIME**, **R**hetorical, and **S**yntactic-aware model for document-level temporal relation extraction. TIMERS uses discourse features in the form of connections from Rhetorical Structure Theory (RST) parsers (Bhatia et al., 2015) to leverage long-range inter-sentential relationships. It also extends existing contextual embeddings with structural and syntactic dependency parse connections. Lastly, it uses timex-timex relations, *dct* (document creation date)-timex relations, and temporal arguments obtained via sentence-level semantic role labeling. These rhetorical, syntactic, and temporal features are learned through a modified version of Relational Graph Convolutional Networks (R-GCN) with a gating mechanism (GR-GCN) (Schlichtkrull et al., 2018), which learns highly relational data relationships in densely-connected graph networks.

Our **main contribution** is a document-level model that incorporates these three features to improve temporal relationship extraction. We obtain state-of-the-art performance across three datasets with **5-18% relative improvement**, showing improvement for events that require chain reasoning, causal prerequisite links, and future events.

2 Methodology

Let document D be defined as a sequence of n tokens $w_i \in W = \{w_1, \cdots, w_n\}$. The entire document is a list of m sentences $V = [v_1, \cdots, v_m]$. Each document has a set of p events $E = \{e_1, \cdots, e_p\}$ and q timexes $T = \{t_1, \cdots, t_q\}$, where $p, q \leq n$. The creation date of the document is represented by timestamp t_{DCT}. We denote the source and target events by e_s and e_t, respec-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 524–533
August 1–6, 2021. ©2021 Association for Computational Linguistics

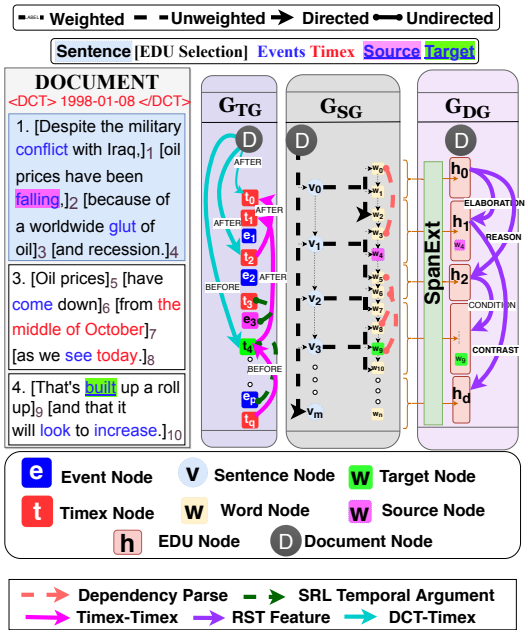


Figure 1: Three graphs are created from the input document. Time-aware Graph (G_{TG}): DCT-Timex associations, Timex-Timex associations, and Temporal Argument connections from semantic role labels; Syntactic-aware Graph (G_{SG}): structural and syntactic connections; and Rhetoric-aware Graph (G_{DG}): rhetorical relations between EDU's (h_i).

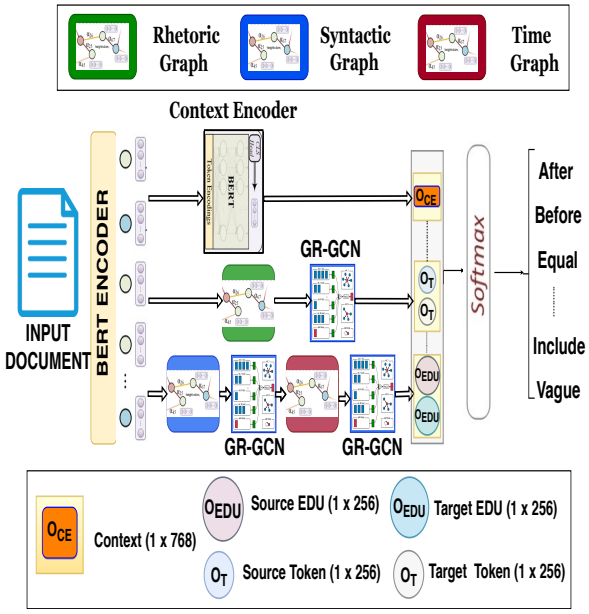


Figure 2: TIMERS learns rhetorical, syntactic, and temporal features through a Gated Relational-Graph Convolutional Networks (GR-GCN). The output of G_{SG} forms the input of G_{TG}. The output corresponding to the source and target nodes learned by G_{TG} (O_T) and G_{DG} (O_{EDU}) are concatenated with the output of the BERT based context encoder (O_{CE}), which forms the final output h_G that passes through the Softmax layer to predict the temporal relation.

tively. The task is to identify the temporal relation $y \in R$ between the source and target event in a multi-class classification setup, where R is the set of all possible temporal links (*TLINKs*).

To solve this task, our model (Fig.1) builds the **TIMERS**-graph, which consists of a Syntactic Graph (Sec.2.1), a Time Graph (Sec. 2.2), and a Rhetorical Graph (Sec.2.3). Each graph is learned through GR-GCN to extract the embeddings used for temporal relation extraction (Fig.2, Sec.2.4).

2.1 Syntactic-Aware Graph

The syntactic graph captures the document structure and word dependency. Our syntactic-aware graph ($\mathcal{G}_{SG}$) is made of separate nodes to represent the document D, each of its inherent sentences $v_i \in V$, and all the constituent words $w_i \in W$ of each sentence. The edges of the Syntactic Graph encode five relations: **(1) Document-Sentence Affiliation** and **(2) Sentence-Word Affiliation** model the hierarchical structure of the document through a directed edge from the document node to each sentence node and from a sentence node to each word in the sentence. **(3) Sentence-Sentence Adjacency** and **(4) Word-Word Adjacency** to preserve sequential ordering for consecutive sentence and word nodes. **(5) Word-Word Dependency** encodes the syntactical nature of the word-level relationships by adding an undirected edge between two word nodes if they share a parent-child relationship in the sentence-level dependency tree.

We use BERT to encode each w_i and obtain sentence embeddings $v_i^{'}$ by averaging the second-to-last hidden layer of BERT for each token. The document vector embedding $D_i^{'}$ was calculated as the average of all sentence embedding ($D_i^{'} = \sum_{i=0}^{m} v_i^{'}$).

2.2 Time-Aware Graph

When events are anchored to a specific time, it becomes easier to infer event relationships from their associated date and time. The time-aware graph ($\mathcal{G}_{TG}$) exploits this intuition and propagates relational information among events, timexes, and the Document Creation Time (*DCT*). The document node D is the node corresponding to the document creation date while the timexes t_i and events e_i are characterized by their corresponding word nodes in the Syntactic Graph. We design three types of edge connections: **(1) *DCT*-Timex Association:** exploit the ordering of timexes with respect to the document creation time through directed weighted edges from *DCT* to timexes. **(2) Timex-Timex Association:** capture inherent non-local timeline ordering between timex pairs by a

directed weighted edge. **(3) Predicate-Temporal Argument:** anchor local temporal relations at the sentence level by connecting each event verb predicate to its temporal argument with a directed edge. The connections formed between temporal entities help navigate information from the source event to the target event while exploring interactions with other events, timexes, *dct*, and temporal arguments.

We calculate timestamps for timexes and the *DCT* from the annotated TimeML format of input documents. The weight of the *DCT*-timex and timex-timex edges is determined based on the temporal order of the entities $\{After, Before, Simultaneous, None\}$. We added *None* as a relation when one of the timestamps cannot be anchored in time.

2.3 Rhetorical-Aware Graph

We use discourse features based on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) to leverage long-range inter-dependencies through a discourse tree. The rhetorical discourse tree of a document contains nodes of phrases, where each phrase (a.k.a, Elementary Discourse Unit or EDU) is contiguous, adjacent and non-overlapping. The interdependencies among EDUs are represented by conventional rhetorical relations (Mann, 1987), e.g. *Elaboration, Span, Condition, Attribution*. Prior work showed discourse features in the form of RST connections help leverage long-range document-level interactions between phrase units (Bhatia et al., 2015) and identify background-foreground events (Aldawsari et al., 2020).

Elementary Discourse Unit (EDU), a sub-sentence phrase unit, is the minimal selection unit for discourse segmentation of a document. We generate the document vector representations at EDU-level $h_i \in H = \{h_1, \cdots, h_d\}$ via the Self-Attentive Span Extractor (SpanExt) from Lee et al. (2017) over the BERT token embeddings. We use the converted dependency version of the tree to build the Rhetorical-aware graph ($\mathcal{G}_{DG}$) by treating every discourse dependency from the i-th EDU to the j-th EDU as a directed edge weighted by the type of the rhetorical relation.

2.4 Temporal Relation Extraction

Each graph is instantiated as a gated variant of Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2018), which we term as Gated Relational Graph Convolution Network (GR-GCN). GR-GCN propagates messages among the nodes to obtain a learned node representation and is inspired by (Zhang et al., 2020). Fig. 2 shows how the learned representations obtained from the syntactic-aware graph forms the input to the time-aware graph. For the time-aware graphs, the learned representations of nodes corresponding to the source event e_s and target event e_t are extracted (O_T). In the case of the rhetorical graphs, the span representations of the EDU span nodes corresponding to the source event (h_e) and target event (h_s) are extracted (O_{EDU}).

The output corresponding to the source and target nodes learnt by G_{TG} (O_T) and G_{DG} (O_{EDU}) are concatenated with output of BERT based context encoder (O_{CE}) (similar to BERT encoding in (Zhao et al., 2020a)): $z_G = \text{ReLU}(W[O_T; O_{EDU}; O_{CE}]+b)$. This is followed by a Softmax layer to predict temporal relations.

Dataset	Train	Validation	Test	Labels
TDDMan (Naik et al., 2019)	4000	650	1500	a, b, s, i, ii
TDDAuto (Naik et al., 2019)	32609	1435	4258	a, b, s, i, ii
MATRES (Ning et al., 2018a) ##	231	25	20	e,a,b,v
TimeBank-Dense (Cassidy et al., 2014)	4032	629	1427	a, b, s, i, ii, v

Table 1: Train/Val/Test data distribution for TDDMan, TDDAuto, MATRES, and TimeBank-Dense; a: After, b: Before, s: Simultaneous, i: Includes, ii: Is_included, v: Vague, e: Equal. (## Ning et al. (2019) use TimeBank and Aquaint for training, Platinum for test; 20% of train as validation)

Corpus	Model	F1
TB-Dense	Vashishtha et al. (2019)	56.6
	EventPlus (Ma et al., 2021)	64.5
	CTRL-PG (Zhou et al., 2020)	65.2
	DEER (Han et al., 2020a)	66.8
	TIMERS (ours)	**67.8**
MATRES	CogCompTime (Ning et al., 2018b)	66.6
	Goyal and Durrett (2019)	68.61
	BiLSTM+MAP (Han et al., 2019c)	75.5
	EventPlus (Ma et al., 2021)	75.5
	Wang et al. (2020)	78.8
	DEER (Han et al., 2020a)	79.3
	Zhao et al. (2020a)	79.6
	SMTL (Ballesteros et al., 2020)	81.6
	TIMERS (ours)	**82.3**

Table 2: Comparison of TIMERS with recent state-of-the-art models on TimeBank-Dense and MATRES dataset. TIMERS outperforms all recent top-performing systems.

3 Experiments

3.1 Data

We train and test our proposed model using the TDDMan and TDDAuto subsets of the TDDiscourse corpus (Naik et al., 2019), which was designed to explicitly focus on global discourse-level temporal ordering. We also train and evaluate our method on the MATRES and TimeBank-Dense datasets, both of which primarily consist of local TLINKs that occur in either the same or adjacent

	System	TDDMan			TDDAuto			MATRES			TB-Dense		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baselines	Majority	37.8	36.3	37.1	34.2	32.3	33.2	50.7	50.7	50.7	40.5	40.5	40.5
	CAEVO (Chambers et al., 2014)	32.3	10.7	16.1	61.1	32.6	42.5	-	-	-	49.9	46.6	48.2
	SP (Ning et al., 2017)	22.7	22.7	22.7	43.2	43.2	43.2	66.0	72.3	69.0	37.7	37.8	37.7
	SP+ILP (Ning et al., 2017)	23.9	23.8	23.8	46.4	45.9	46.1	71.3	82.1	76.3	58.4	58.4	58.4
	BiLSTM (Cheng and Miyao, 2017)	24.9	23.8	24.3	55.7	48.3	51.8	59.5	59.5	59.5	**63.9**	38.9	48.4
	BERT-base Transformer	36.5	37.1	37.5	62.0	61.7	62.3	65.6	78.1	77.2	59.7	60.7	62.2
	RoBERTa-base	35.7	36.5	37.1	60.6	62.7	61.6	77.3	79.0	78.9	58.1	57.6	61.9
	TIMERS (ours)	**43.7***	**46.7***	**45.5***	**64.3***	**72.7***	**71.1***	**81.1***	**84.6***	**82.3***	48.1	**65.2***	**67.8**
Ablation	TIMERS w\o Context Encoder	29.7	35.5	33.7	49.8	52.5	51.6	61.2	69.6	68.6	43.8	54.5	50.6
	TIMERS w\o $\mathcal{G}_{DG}$	39.6	39.6	41.8	61.7	66.8	65.4	71.8	79.1	79.7	51.4	63.0	63.3
	TIMERS w\o $\mathcal{G}_{SG}$	38.5	42.6	42.3	63.3	69.5	68.9	71.6	78.5	78.2	51.1	62.1	62.8
	TIMERS w\o $\mathcal{G}_{TG}$	37.5	39.8	39.5	58.7	68.3	67.1	72.8	78.5	77.7	50.5	62.9	61.8

Table 3: Results comparing performance of TIMERS with baselines and ablative components on TDDMan, TDDAuto, MATRES and TimeBank-Dense datasets. We adopt the BERT and RoBERTa implementation from (Ballesteros et al., 2020). * indicates statistical significance over BERT Transformer ($p \leq 0.005$) under Wilcoxon's Signed Rank test. Darker green represents better F1 performance on ablation studies. Bold denotes the best performing model. TIMERS improves substantially over all datasets. The ablation shows that context, discourse ($\mathcal{G}_{DG}$), and time-aware ($\mathcal{G}_{TG}$) graph encoders prove to be most beneficial.

sentences. Table 1 reports the data statistics and label distributions. (Naik et al., 2019) shows the distribution of the distance between event-pairs for all TLINKs in the TDD test set and explains that nearly 53% TLINKs in the TDD dataset comprise of event pairs that are more than 5 sentences apart. Like Cheng and Miyao (2017), we report results on non-vague labels of TimeBank-Dense. MATRES has no standard validation set. Hence, we follow the split used in (Ning et al., 2019).

3.2 Experimental Settings

Token Encoding:The word-level token representations are obtained by summing the corresponding BERT embeddings from the last 4 layers of pre-trained BERT-base encoder. **Syntactic Dependency Parser**: The dependency parse tree of individual sentences is obtained via SpaCy[1] to form word-word dependency connections in the syntactic-aware graph. **Semantic Role Labeller**: We extract semantic role labels using AllenNLP's SRL parser[2] that internally uses SRL-BERT (Shi and Lin, 2019) to obtain the temporal arguments corresponding to each verb event. **Timex Normalization**: Timex phrases are treated as a single unit for the purpose of graph construction by average pooling their BERT tokenized representations. Microsoft Recognizers-Text[3] is employed to normalize timexes and DCT date-time values. The normalized timex expressions are compared through Allen's interval algebra, where each timex has a start and an endpoint. The comparison is then made on the basis of the endpoints of the timexes, forming an edge going from earlier to later ending timex. **RST Discourse Parser**: We used the shift-reduce discourse parser proposed by Ji and Eisenstein (2014) to build the discourse tree [4], which is post-processed using *discoursegraphs* library[5] (Neumann, 2015) to build the rhetorical dependencies graph. Further implementation details can be found in the appendix.

3.3 Results

Table 3 compares our work to the baseline methods reported on the TDDMan, TDDAuto, MATRES, and TimeBank-Dense datasets. We also include results for BERT-based Transformer (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) following Ballesteros et al. (2020). To prevent truncation or memory errors otherwise caused by multi-sentence spans, we concatenate only sentences containing source and events as input to Transformer baselines. These methods outperform the existing reported results and provide strong benchmarks but still perform similarly to a majority class baseline for the TDDMan dataset. Our model shows a significant gain of 8.0 F1 and 8.8 F1 over the BERT baseline on the TDDMan and TDDAuto datasets. Table 2 compares TIMERS to additional rigorous state-of-the-art methods for TimeBank-Dense and MATRES. TIMERS achieves state-of-the-art performance on all four datasets, showing that it successfully handles intra-sentence, inter-sentence, and cross-sentence TLINK pairs through the same architecture.

[1] https://spacy.io/
[2] https://demo.allennlp.org/semantic-role-labeling
[3] https://github.com/microsoft/Recognizers-Text
[4] Implementation used: https://github.com/jiyfeng/DPLP
[5] https://pypi.org/project/discoursegraphs/

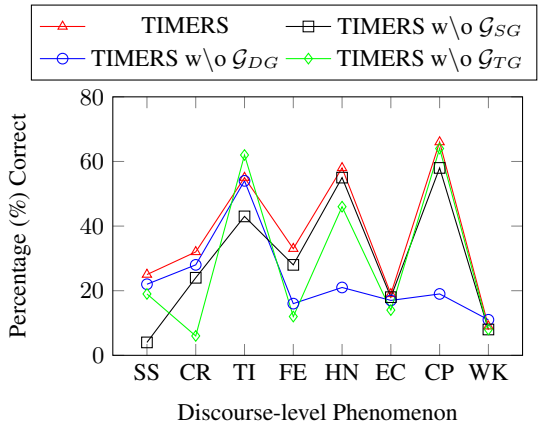


Figure 3: Error analysis on manually annotated discourse-level phenomena in the test set of TDDMan. SS: Single-Sent, CR: Chain Reasoning, TI: Tense Indicator, FE: Future Events, HN: Hypothetical/Negated, EC: Event Coreference, CP: Causal/Prereq, WK: World Knowledge. TIMERS handles CR and CP phenomena but struggles on EC and WK.

3.4 Ablation Study

To assess the contribution of discourse, syntactic, and time-aware graphs, we performed an ablation experiment with different configurations (Table 3). Removing the context encoder significantly degrades performance, indicating that the graph components themselves cannot replace the contextual encoding. Removing any of the graph encoders hurts the model performance, motivating the need for all the constituent graph components. We also analyzed the relative importance of $\mathcal{G}_{DG}$, $\mathcal{G}_{SG}$, and $\mathcal{G}_{TG}$ represented by color shading in the table. The results show that the syntactic graph is least important for document level pairs in TDDMan and TDDAuto, which we believe is due to the longer range dependencies present in this dataset. However, removing the discourse graph for TimeBank-Dense and MATRES datasets leads to the least performance deterioration as inter and intra-sentence pairs do not fully utilize document-level rhetorical relations. TIMERS outperforms the BERT baseline even without $\mathcal{G}_{TG}$, demonstrating its useful in cases where document creation date or timexes cannot be obtained easily.

3.5 Error Analysis

The error analysis results of TIMERS and its ablations for TDDMan are shown in Fig. 3 (the results on TDDAuto are in Appendix Fig.1). The results provide evidence that the syntactic-aware graph ($\mathcal{G}_{SG}$) is most important for relations that can be extracted from a single sentence (SE). The time-aware graph ($\mathcal{G}_{TG}$) plays an important role in improving relationships requiring chain reasoning (multi-hop) and relationship determined by future events. We also note the role of the rhetorical-aware graph ($\mathcal{G}_{DG}$) for modeling future possibility (FE), hypothetical events (HN) and causal conditions for event occurrences (CP). This can be attributed to rhetorical relational features that extract plausible inter-dependencies such as *cause, explanation, contrast* (Lioma et al., 2012). None of the experimented models show improved performance on TLINK pairs which depend on world knowledge (WK) or event coreference (EC).

4 Conclusion

This work presents a neural architecture that utilizes local syntactic features, rhetorical discourse features, and temporal arguments in semantic role labels through a Gated Relational-GCN for document-level temporal relation extraction on TDDiscourse, MATRES, and TimeBank-Dense datasets. Experiments show that TIMERS shows substantial improvement for events that require chain reasoning and causal prerequisite links. Future work will focus on exploring real-world scenarios in which the temporal extraction task suffers from absent or erroneous event and timex annotations. We believe our proposed methods can also be adapted for other languages as well by overcoming possible limitations such as dependency parsing, semantic parsing, Timex normalization for the non-English corpora.

Ethics Statement

This work does not collect or release any new data resource. Moreover, all four of the datasets used in experiments (TDDiscourse, TimeBank-Dense and MATRES) are publicly available and free to use, hence do not intrude user privacy. During the course of this work, no human judgements were exploited nor any user-level data was collected, stored or processed. Our methods do not add to any pre-existing data biases. Potential applications of this work include extracting event timelines from news, contractual documents, and digitizing patient electronic health records. We acknowledge that temporal information extraction finds applications in clinical NLP (Lin et al., 2019; Tourille et al., 2017). Hence, we would like to caution about shortcomings of the proposed system in terms of misclassifications on event pairs requiring real-world common sense reasoning and domain shift.

References

Mohammed Aldawsari, Adrian Perez, Deya Banisakher, and Mark Finlayson. 2020. Distinguishing between foreground and background events in news. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5171–5180, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Miguel Ballesteros, Rishita Anubhai, Shuai Wang, Nima Pourdamghani, Yogarshi Vyas, Jie Ma, Parminder Bhatia, Kathleen McKeown, and Yaser Al-Onaizan. 2020. Severing the edge between before and after: Neural architectures for temporal ordering of events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5412–5417, Online. Association for Computational Linguistics.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218, Lisbon, Portugal. Association for Computational Linguistics.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2019. Embedding time expressions for deep temporal ordering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4400–4406, Florence, Italy. Association for Computational Linguistics.

Rujun Han, I-Hung Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. 2019a. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 666–106, Hong Kong, China. Association for Computational Linguistics.

Rujun Han, Mengyue Liang, Bashar Alhafni, and Nanyun Peng. 2019b. Contextualized word embeddings enhanced event temporal relation extraction for story understanding. *ArXiv*, abs/1904.11942.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019c. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China. Association for Computational Linguistics.

Rujun Han, X. Ren, and Nanyun Peng. 2020a. Deer: A data efficient language model for event temporal reasoning. *ArXiv*, abs/2012.15283.

Rujun Han, Yichao Zhou, and Nanyun Peng. 2020b. Domain knowledge empowered structured neural net for end-to-end event temporal relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5717–5729, Online. Association for Computational Linguistics.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

A. Leeuwenberg and Marie-Francine Moens. 2019. A survey on temporal reasoning for temporal information extraction from text. *ArXiv*, abs/2005.06527.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246, Brussels, Belgium. Association for Computational Linguistics.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language*

Processing Workshop, pages 65–71, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Christina Lioma, Birger Larsen, and Wei Lu. 2012. Rhetorical relations for information retrieval. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 931–940. ACM.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Mingyu Derek Ma, J. Sun, M. Yang, Kung-Hsiang Huang, N. Wen, Shikhar Singh, Rujun Han, and Nanyun Peng. 2021. Eventplus: A temporal event understanding pipeline. *ArXiv*, abs/2101.04922.

W. Mann. 1987. Rhetorical structure theory: A theory of text organization.

W. Mann and S. A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text Talk*, 8:243 – 281.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an LSTM-based architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 887–896, Copenhagen, Denmark. Association for Computational Linguistics.

Aakanksha Naik, Luke Breitfeller, and Carolyn Rose. 2019. TDDiscourse: A dataset for discourse-level temporal ordering of events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 239–249, Stockholm, Sweden. Association for Computational Linguistics.

Arne Neumann. 2015. discoursegraphs: A graph-based merging tool and converter for multilayer annotated corpora. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 309–312, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037, Copenhagen, Denmark. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018a. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018b. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77, Brussels, Belgium. Association for Computational Linguistics.

Yunseok Noh, Yongmin Shin, Junmo Park, A.-Yeong Kim, Su-Jeong Choi, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2020. WIRE: an automated report generation system using topical and temporal summarization. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2169–2172. ACM.

J. Pustejovsky, José M. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, and Dragomir R. Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the TimeBank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2195–2204, Berlin, Germany. Association for Computational Linguistics.

M. Schlichtkrull, Thomas Kipf, P. Bloem, R. V. Berg, Ivan Titov, and M. Welling. 2018. Modeling relational data with graph convolutional networks. *ArXiv*, abs/1703.06103.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Julien Tourille, Olivier Ferret, Aurélie Névéol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–230, Vancouver, Canada. Association for Computational Linguistics.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 696–706, Online. Association for Computational Linguistics.

Zhenyu Zhang, Bowen Yu, Xiaobo Shu, Tingwen Liu, Hengzhu Tang, Wang Yubin, and Li Guo. 2020. Document-level relation extraction with dual-tier heterogeneous graph. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1630–1641, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xinyu Zhao, Shih ting Lin, and Greg Durrett. 2020a. Effective distant supervision for temporal relation extraction. *ArXiv*, abs/2010.12755.

Xinyu Zhao, Shih-ting Lin, and Greg Durrett. 2020b. Effective distant supervision for temporal relation extraction. *arXiv preprint arXiv:2010.12755*.

Yichao Zhou, Yu Yan, Rujun Han, J. Caufield, Kai-Wei Chang, Y. Sun, P. Ping, and W. Wang. 2020. Clinical temporal relation extraction with probabilistic soft logic regularization and global inference. *ArXiv*, abs/2012.08790.

A Experiment Settings

A.1 Node Connections

We detail the node connections present in each graph of our proposed model along with edge attributes in Table 4.

A.2 Edge Relations

Table 6 lists rhetorical relations used in Rhetoric-aware graph G_{DG}in the TIMERS model, along with the definitions as provided by Mann (1987). The weights of the Rhetoric graph G_{DG} are determined based on the RST relations described in this table. Table 7 details the type of relations between timex-timex and DCT-timex nodes of the Time-aware graph G_{TG}.

A.3 Training Setup

Hyperparameter: Hyper-parameters for our model were tuned on the respective validation set to find the best configurations for different datasets. We summarize the range of our model's hyper parameters such as: number of hidden layers in GR-GCN $\{1, 2, 3\}$, size of hidden layers in GR-GCN $\{64, 128, 256, 512\}$, BERT embedding size, dropout $\delta \in \{0.2, 0.3, 0.4, 0.5.0.6\}$, learning rate $\lambda \in \{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$, weight decay $\omega \in \{1e-6, 1e-5, 1e-4, 1e-3\}$, batch size $b \in \{16, 32, 64\}$ and epochs (≤ 100).

Contextual Encoder: We used BERT-base-uncased for generating token embedding of size 1x 768. As BERT-base Transformer provides a stronger baseline as compared to RoBERTa, we utilized BERT Transformer for Contextual Encoder in TIMERS architecture. We use the default dropout rate (0.1) on BERT's self attention layers but do not use additional dropout at the top linear layer The output from the Contextual Encoder is a 1-D vector of size 768.

Loss Function and Inference: TIMERS is trained end to end using Binary Cross Entropy loss with Adam optimizer. Across all four datasets, we found the best results correspond with the use of Adam optimiser set with default values $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$, weight-decay of $5e-4$ and an initial learning rate of 0.001. We evaluate the performance of temporal relation extraction systems in terms of F1, precision and recall score.

Computing Infrastructue: TIMERS is written in PyTorch library and was trained on Nvidia GeForce RTX 2080 GPU. **Average Runtime**: The model takes a maximum of approximately 6,500 seconds to train on either of the four datasets.

Dataset Access Links to download TDDiscourse (Naik et al., 2019) dataset: https://github.com/aakanksha19/TDDiscourse Link to download MATRES (Ning et al., 2018a) dataset: https://github.com/qiangning/MATRES Link to download TimeBank-Dense (Cassidy et al., 2014) dataset: https://github.com/muk343/TimeBank-dense

A.4 Reproducibility

Table 5 lists the range ad best values of the hyperparameters used in TIMERS model for different data settings. We used grid search to choose the best set of training configurations across each dataset. We run 5 rounds of hyper-parameter search trials and report average of observed results.

B Additional Results

We observe from Figure 4 a similar trend to TDDMan, although with a stronger support for SS, CR, TI and and FE. This is partly due to the fact that TDDAuto was generated automatically (Naik et al., 2019) using weakly annotated time relations. Moreover, 90% of samples in TDDAuto require SS. Hence, TIMERS trained exclusively on TDDAuto performs worse on challenging phenomenon like HN and CP. Consistent with results on TDDMan, TIMERS and its ablations trained on TDDAuto struggle on EC and WK.

Edge	Graph	Source	Target	Directed	Weighted
Document-Sentence Affiliation	Syntactic	Doc Node	Sent Nod	✓	✗
Sentence-Word Affiliation	Syntactic	Sent Nod	Word Node	✓	✗
Sentence-Sentence Adjacency	Syntactic	Sent Nod	Sent Nod	✓	✗
Word-Word Adjacency	Syntactic	Word Node	Word Node	✓	✗
Word-Word Dependency	Syntactic	Word Node	Word Node	✗	✗
DCT-Timex Association	Time	Doc Node	Timex	✓	✓
Timex-Timex Association	Time	Timex	Timex	✓	✓
Predicate-Temporal Argument	Time	Word Node	Timex	✗	✗
RST Discourse	Discourse	EDU	EDU	✓	✓

Table 4: List of node connections in TIMERS.

	Dataset			
Hyperparameters	TDDMan	TDDAuto	MATRES	TB-Dense
Dropout Ratio	0.5	0.5	0.5	0.5
Optimizer	Adam	Adam	Adam	Adam
Input Dimension (Context Encoder)	(n,768)	(n,768)	(n,768)	(n,768)
Input Dimension (Syntactic Graph)	(n,768)	(n,768)	(n,768)	(n,768)
Input Dimension (Time Graph)	(n,256)	(n,256)	(n,64)	(n,64)
Input Dimension (Rhetoric Graph)	(n,768)	(n,768)	(n,768)	(n,768)
Hidden Dimension (GR-GCN)	256	256	64	64
Number of hidden layers (GR-GCN)	1	1	1	1
Hidden Dimension of SpanExt	{256, 64}	{256, 64}	{128, 64}	{128, 64}
Epochs	20	20	20	20
Batch Size	8	8	16	16
Activation Function of Linear layers	ReLU	ReLU	ReLU	ReLU
Dimension of final FCN	[(1792 x r)]	[(1792 x r)]	[(1024 x r)]	[(1024 x r)]
Output Classes	5	5	4	5

Table 5: **Hyperparameters Details:** Training hyperparameters of TIMERS for TDDMan, TDDAuto, MATRES and TB-Dense datasets. n refers to the number of input samples; r refers to the number of total relation classes

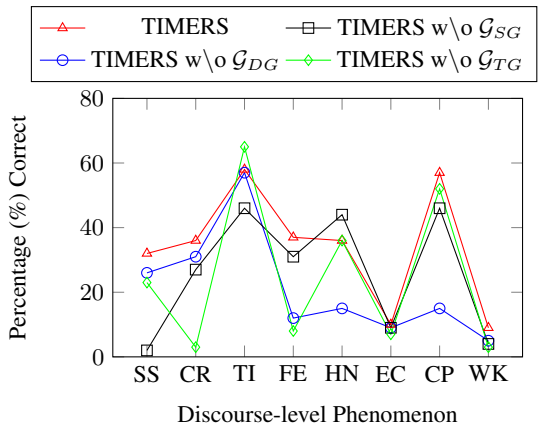


Figure 4: Error analysis on manually annotated discourse-level phenomenon in test set of TDDAuto. SS: SingleSent, CR: Chain Reasoning, TI: Tense Indicator, FE: Future Events, HN: Hypothetical/Negated, EC: Event Coreference, CP: Causal/Prereq, WK: World Knowledge. We observe a stronger support for SS, CR, TI and and FE as compared to TDDMan. TIMERS trained exclusively on TDDAuto performs worse on challenging phenomenon like HN and CP. Consistent with results on TDDMan, TIMERS and its ablations trained on TDDAuto struggle on EC and WK.

Relation Label	Definition
Temporal	Relating to time
Summary	Shorter restatement
Same-unit	Part of the same phrasal unit
Span	Extending to multiple phrasal units
Purpose	Initiation in order to realize a goal
Example	Specific subtypes
Elaboration	Providing additional details
Reason	Justification with intent to defend a stance
Sequence	Subject-matter sequence
Condition	Realization of dependency
Means	Method or instrument to improve likelihood
Consequence	Intended or unintended end goal
Topic	Central idea
Attribution	Contributing factor
Textual Organization	Part of formal text span
Contrast	Opposing phenomenon
Manner	Semantic course of occurrence
Antithesis	Incompatibility due to contrast
Concession	Potential Incompatibility
Explanation	Providing clarification to an established fact
Circumstance	Framework for interpretation

Table 6: RST relations used in Rhetoric-aware graph G_{DG} in TIMERS, with definition as provided by Mann (1987)

Relation Label	Definition
After	TIMEX1 starts after TIMEX2 has ended
Before	TIMEX1 ends before TIMEX2 started
Equal	TIMEX1 is numerically equal to TIMEX2 upto date resolution.
None	One of the timex cannot be extracted or normalized

Table 7: Timex-Timex and DCT-Timex relations used in the Time-aware graph G_{TG}.

Improving Arabic Diacritization with Regularized Decoding and Adversarial Training

Han Qin[♠*], **Guimin Chen**[♢*], **Yuanhe Tian**[♥*], **Yan Song**[♠♡†]
[♠]The Chinese University of Hong Kong (Shenzhen)
[♢]QTrade [♥]University of Washington
[♡]Shenzhen Research Institute of Big Data
[♠]hanqin@link.cuhk.edu.cn [♢]chenguimin@foxmail.com
[♥]yhtian@uw.edu [♠]songyan@cuhk.edu.cn

Abstract

Arabic diacritization is a fundamental task for Arabic language processing. Previous studies have demonstrated that automatically generated knowledge can be helpful to this task. However, these studies regard the auto-generated knowledge instances as gold references, which limits their effectiveness since such knowledge is not always accurate and inferior instances can lead to incorrect predictions. In this paper, we propose to use regularized decoding and adversarial training to appropriately learn from such noisy knowledge for diacritization. Experimental results on two benchmark datasets show that, even with quite flawed auto-generated knowledge, our model can still learn adequate diacritics and outperform all previous studies, on both datasets.[1]

1 Introduction

Modern standard Arabic (MSA) is generally written without diacritics, which poses a challenge to text processing and understanding in downstream applications, such as text-to-speech generation (Drago et al., 2008) and reading comprehension (Hermena et al., 2015). Restoration of such diacritics, known as diacritization, becomes an important task for Arabic natural language processing (NLP). Among different diacritization methods (Pasha et al., 2014; Shahrour et al., 2015; Zitouni et al., 2006; Habash and Rambow, 2007; Darwish et al., 2017), the neural ones (Abandah et al., 2015a; Fadel et al., 2019a,b; Zalmout and Habash, 2019, 2020; Darwish et al., 2020) achieve the best performance due to their better capability in incorporating contextual features. To further improve diacritization, automatically generated knowledge from off-the-shelf toolkits, such as morphological features, parts-of-speech tags, and automatic diacritization results, have been extensively applied to this task (Zitouni et al., 2006; Arabiyat, 2015; Darwish et al., 2017, 2020). However, current models treat such knowledge instances as gold references and always directly concatenate them with input embeddings (Arabiyat, 2015; Darwish et al., 2020), which may lead to inferior results since the knowledge may be inaccurate, especially if the toolkits were trained on data with different criteria.

Diacritization can be performed by character-based sequence labeling (Zitouni et al., 2006; Belinkov and Glass, 2015; Fadel et al., 2019b). We follow this paradigm and propose a neural approach in this paper, using regularized decoding and adversarial training to incorporate auto-generated knowledge (i.e., the diacritization results generated from off-the-shelf toolkits). Specifically, the regularized decoder treats the auto-generated knowledge as separate gold labels and learns to predict them in a separate decoding process, which is then used to update the main model. The adversarial training is applied to the encoding process by determining whether the diacritization for an input follows the gold label or the auto-generated knowledge. In doing so, our model can dynamically distinguish between auto-generated knowledge instances instead of treating them all as gold references, so as to effectively identify what knowledge should be leveraged for different inputs. Importantly, regularized decoding and adversarial training are exclusively applied to the training stage; we only need the main tagger for inference once the model has been trained. Experimental results and further analyses illustrate the effectiveness of our approach, where our model outperforms strong baselines and achieves state-of-the-art results on two benchmark datasets: Arabic Treebank (ATB) (Maamouri et al., 2004) and Tashkeela (Zerrouki and Balla, 2017).

[*]Equal contribution.
[†]Corresponding author.
[1]The code and models involved in this paper are released at https://github.com/cuhksz-nlp/AD-RDAT.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 534–542
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 The Proposed Approach

As shown in Figure 1, our approach for diacritization follows the sequence labeling paradigm, where it has two training stages for the main tagger ($\mathcal{M}$). In the **first training stage** (presented in the orange box in Figure 1), $\mathcal{M}$ is enhanced by regularized decoding ($\mathcal{RD}$) and adversarial training ($\mathcal{AT}$) to discriminatively learn from the auto-generated labels. Specifically, given an input Arabic character sequence $\mathcal{X} = x_1 \cdots x_i \cdots x_n$, $\mathcal{M}$ and $\mathcal{RD}$ aim to predict two types of diacritization labels, $\widehat{\mathcal{Y}}$ and $\widehat{\mathcal{Y}}^K$, which follow the gold and auto-generated label criteria, respectively. $\mathcal{AT}$ ensures that the main tagger only learns useful information from either gold or auto-generated labels. Therefore, the first training stage can be conceptually formalized by

$$\widehat{\mathcal{Y}}, \widehat{\mathcal{Y}}^K = f(\mathcal{M}(\mathbf{H}^S, \mathcal{X}), \mathcal{RD}(\mathbf{H}^S, \mathcal{X}), \mathcal{AT}(\mathbf{H}^S)) \quad (1)$$

where $\mathbf{H}^S$ denotes the output vectors of the shared encoder $\mathcal{SE}$ (whose input is $\mathcal{X}$) that is designed to learn the information shared by the gold and auto-generated labels. As a result, the goal of this training stage is to minimize the loss defined by

$$\mathcal{L} = \mathcal{L}_\mathcal{M} + \mathcal{L}_\mathcal{K} + \mathcal{L}_\mathcal{A} \quad (2)$$

where $\mathcal{L}_\mathcal{M}$, $\mathcal{L}_\mathcal{K}$ and $\mathcal{L}_\mathcal{A}$ refer to the losses that come from $\mathcal{M}$, $\mathcal{RD}$, and $\mathcal{AT}$, respectively.

Afterwards, in the **second training stage** (presented in the green box in Figure 1), $\mathcal{M}$ is further trained alone on the gold labels $\widehat{\mathcal{Y}}$ without using auto-generated $\widehat{\mathcal{Y}}^K$, $\mathcal{RD}$ and $\mathcal{AT}$, to fine-tune its parameters, where all parameters in $\mathcal{SE}$ obtained through the first training stage are fixed. For inference, only $\mathcal{M}$ is used without requiring any additional input other than $\mathcal{X}$ to obtain the diacritization results. In the following sections, we first describe $\mathcal{M}$, then elaborate the details of $\mathcal{RD}$ and $\mathcal{AT}$.

2.1 The Main Tagger

The main tagger uses an encoder-decoder architecture, as shown in Figure 1, in which a shared encoder $\mathcal{SE}$ and a private encoder $\mathcal{PE}^\mathcal{M}$ are applied to model the contextual information. Particularly, $\mathcal{SE}$ is proposed to facilitate the process of leveraging auto-generated knowledge, which is excepted to learn information shared by the gold labels and the auto-generated knowledge. It takes the character embeddings of $\mathcal{X}$ (the embedding of x_i is denoted as $\mathbf{e}_i$) as input and encodes them to the shared hidden vectors (denoted as $\mathbf{h}_i^S$ for x_i) by

$$[\mathbf{h}_1^S, \cdots, \mathbf{h}_n^S] = \mathcal{SE}([\mathbf{e}_1, \cdots, \mathbf{e}_n]) \quad (3)$$

Similarly, $\mathcal{PE}^\mathcal{M}$ is also applied to the word embeddings and produces the result $\mathbf{h}_i^M$. Then, we concatenate $\mathbf{h}_i^S$ and $\mathbf{h}_i^M$ and map the resulting vector to the output space with a fully connected layer: $\mathbf{o}_i^M = \mathbf{W}^M \cdot (\mathbf{h}_i^M \oplus \mathbf{h}_i^S) + \mathbf{b}^M$, where $\oplus$ is concatenation and $\mathbf{W}^M$ and $\mathbf{b}^M$ are the trainable matrices and bias vector, respectively. Finally, a *softmax* decoder is applied to $\mathbf{o}_i^M$ to predict the label $\widehat{y}_i$:

$$\widehat{y}_i = \arg\max \frac{exp(o_i^{M,t})}{\sum_{t=1}^{|\mathcal{T}|} exp(o_i^{M,t})}, \quad (4)$$

where $\mathcal{T}$ denotes the set of all diacritization labels and $o_i^{M,t}$ is the value at dimension t in $\mathbf{o}_i^M$. Therefore the loss for $\mathcal{M}$ is

$$\mathcal{L}_\mathcal{M} = -\sum_{i=1}^{n} \log p(y_i^* | \mathcal{X}), \quad (5)$$

where $p(y_i^* | \mathcal{X})$ denotes the probability of labeling x_i by the gold label y_i^*.

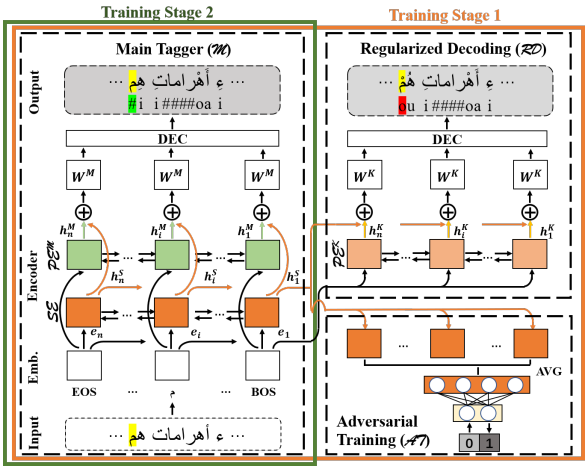


Figure 1: The architecture of our model, where the left shows the main tagger ($\mathcal{M}$) and the right shows the regularized decoding ($\mathcal{RD}$) and adversarial training ($\mathcal{AT}$) modules. The diacritization labels for an example following different criteria are illustrated in $\mathcal{M}$ and $\mathcal{RD}$, with the mismatching labels marked in green and red. E.g., for "م" (highlighted in yellow), its gold and auto-generated labels are "#" (*null*) and "o" (*sukun*).[2]

2.2 Regularized Decoding

When leveraging auto-generated knowledge, it is important to note that such knowledge may be inaccurate or follow different annotation criteria, which is required to be appropriately addressed to pre-

[2]We use a set of symbols to label different diacritization results, which are illustrated in Appendix A.

	ATB					Tashkeela				
	w/ case ending		w/o case ending		ACC	w/ case ending		w/o case ending		ACC
	DER	WER	DER	WER		DER	WER	DER	WER	
BiLSTM	2.28	6.62	1.98	4.15	93.38	2.59	7.62	2.30	5.01	92.98
+$\mathcal{RD}$	2.12	5.90	1.72	3.37	94.10	2.18	6.42	1.87	4.04	94.09
+$\mathcal{RD}$+$\mathcal{AT}$	1.87	5.17	1.59	3.09	94.83	2.10	6.08	1.82	3.88	94.40
Transformer	2.22	6.36	1.92	4.00	93.64	2.70	7.98	2.43	5.37	92.65
+$\mathcal{RD}$	2.07	5.90	1.70	3.45	94.10	2.11	6.11	1.82	3.85	94.37
+$\mathcal{RD}$+$\mathcal{AT}$	**1.83**	**5.09**	**1.56**	**3.07**	**94.91**	**2.06**	**5.98**	**1.76**	**3.75**	**94.49**

(a) AraBERT

	ATB					Tashkeela				
	w/ case ending		w/o case ending		ACC	w/ case ending		w/o case ending		ACC
	DER	WER	DER	WER		DER	WER	DER	WER	
BiLSTM	2.15	6.16	1.81	3.73	93.84	2.48	7.33	2.19	4.79	93.27
+$\mathcal{RD}$	1.97	5.65	1.69	3.48	94.35	2.08	6.02	1.83	3.91	94.50
+$\mathcal{RD}$+$\mathcal{AT}$	1.81	5.06	1.53	3.02	94.94	2.03	5.86	1.77	3.75	94.69
Transformer	2.05	5.80	1.77	3.61	94.20	2.66	7.65	2.33	4.99	92.95
+$\mathcal{RD}$	1.85	5.11	1.56	3.02	94.89	1.96	5.62	1.67	**3.37**	94.86
+$\mathcal{RD}$+$\mathcal{AT}$	**1.77**	**4.88**	**1.49**	**3.01**	**95.12**	**1.87**	**5.54**	**1.56**	3.64	**94.94**

(b) ZEN 2.0

Table 1: Experimental results (i.e., DER and WER with and without the case ending being considered and accuracy) of baselines and our models with $\mathcal{RD}$ and $\mathcal{AT}$ using AraBERT (a) and ZEN 2.0 (b) on the test sets of ATB and Tashkeela, "BiLSTM" and "Transformer" denote the encoders (i.e., $\mathcal{SE}$ and $\mathcal{PE}$) used in the models.

vent the noise in the auto-generated knowledge from significantly hurting the model performance (Tang et al., 2020; Nie et al., 2020; Chen et al., 2020; Mandya et al., 2020; Tian et al., 2020a,b, 2021a,b; Chen et al., 2021). To tackle this challenge, we propose to learn from a special decoding process, which is integrated into the main diacritization model, in order to reduce error propagation compared to directly using the knowledge instances or their features. As shown in Figure 1, the proposed regularized decoding is an extra output process separated from the main tagger and performed on another sequence of labels $\mathcal{Y}^{K^*}$, which are the auto-generated knowledge instances (diacritization labels) annotated by an existing toolkit. Therefore, the loss $\mathcal{L}_\mathcal{K}$ from $\mathcal{RD}$ is computed through

$$\mathcal{L}_\mathcal{K} = -\sum_{i=1}^{n} \log p(y_i^{K^*}|\mathcal{X}) \quad (6)$$

and in the first training stage, all trainable parameters in $\mathcal{SE}$ are updated through the information back-propagated from $\mathcal{RD}$.

2.3 Adversarial Training

Although auto-generated knowledge can be back-propagated through $\mathcal{RD}$, it could be overwhelmed by the information directly learned from the gold label. We further improve our model by balancing the information learned from both $\mathcal{M}$ and $\mathcal{RD}$ with $\mathcal{AT}$, which is proposed to equalize both sides and emphasize the shared information from them.[3] In doing so, we connect a discriminator, which is a binary classifier, to $\mathcal{SE}$. The discriminator takes all $\mathbf{h}_i^S$ from $\mathcal{SE}$, averages them by $\mathbf{h}^S = \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i^S$, and then passes the resulting vector to a fully connected layer with a *softmax* function to compute its bias towards either type (i.e., the gold or auto-generated) of diacritization labels:

$$[p_m, p_k] = softmax(\mathbf{W}^D \cdot \mathbf{h}^S + \mathbf{b}^D) \quad (7)$$

where $\mathbf{W}^D$ and $\mathbf{b}^D$ are the trainable matrix and bias vector, respectively, that map $\mathbf{h}^S$ to a two-dimensional vector, with p_m and p_k representing normalized probabilities that satisfy $p_m + p_k = 1$ and indicating the bias of $\mathcal{SE}$ towards gold and auto-generated labels, respectively. Then we apply a negative log-likelihood loss function to the discriminator, formalized as

$$\mathcal{L}_\mathcal{D} = -\log p_m - \log p_k \quad (8)$$

and an adversarial loss to the parameters in $\mathcal{SE}$ via

$$\mathcal{L}_\mathcal{S} = -p_m \log p_m - p_k \log p_k \quad (9)$$

As a result, the goal of $\mathcal{AT}$ is to minimize the loss

$$\mathcal{L}_\mathcal{A} = \mathcal{L}_\mathcal{D} - \lambda\mathcal{L}_\mathcal{S} \quad (10)$$

[3] $\mathcal{AT}$ follows the idea that the $\mathcal{SE}$ should have no bias towards the information learned from $\mathcal{M}$ and $\mathcal{RD}$.

	ATB					Tashkeela				
	w/ case ending		w/o case ending		ACC	w/ case ending		w/o case ending		ACC
	DER	WER	DER	WER		DER	WER	DER	WER	
Fadel et al. (2019a)	-	-	-	-	-	3.73	11.19	2.88	6.53	-
Abandah and Abdel-Karim (2019)	2.46	8.12	1.24	3.81	-	1.97	5.13	1.22	3.13	-
Fadel et al. (2019b)	-	-	-	-	-	2.60	7.69	2.11	4.57	-
Alqahtani et al. (2019)	2.80	8.20	-	-	-	-	-	-	-	-
Alqahtani et al. (2020)	2.54	7.51	-	-	-	-	-	-	-	-
Zalmout and Habash (2020)	-	-	-	-	93.90	-	-	-	-	-
Farasa	19.84	68.61	20.31	68.48	31.39	22.00	58.66	24.89	53.14	45.96
Ours (AraBERT) (BiLSTM)	1.87	5.17	1.59	3.09	94.83	2.10	6.08	1.82	3.88	94.40
Ours (AraBERT) (Transformer)	**1.83**	**5.09**	**1.56**	**3.07**	**94.91**	**2.06**	**5.98**	**1.76**	**3.75**	**94.49**
Ours (ZEN 2.0) (BiLSTM)	1.81	5.06	1.53	3.02	94.94	2.03	5.86	1.77	3.75	94.69
Ours (ZEN 2.0) (Transformer)	**1.77**	**4.88**	**1.49**	**3.01**	**95.12**	**1.87**	**5.54**	**1.56**	**3.64**	**94.94**

Table 2: Comparisons of experimental results (i.e., DER, WER, and accuracy) between previous studies and our models with AraBERT and ZEN 2.0 embeddings on the test sets of the ATB and Tashkeela.

where λ is a positive coefficient that controls the influence of $\mathcal{L}_{\mathcal{S}}$ in the adversarial training, so that to minimize $\mathcal{L}_{\mathcal{D}}$ and maximize $\mathcal{L}_{\mathcal{S}}$ synchronously.

3 Experiments

3.1 Settings

In our experiments, We use two benchmark datasets, i.e., ATB (Arabic Treebank Part 1, 2, and 3) (Maamouri et al., 2004) and Tashkeela (Zerrouki and Balla, 2017), following the same settings in previous studies.[4] For implementation, we run Farasa[5] (Abdelali et al., 2016) on the two datasets and collect their diacritization results for regularized decoding. Since the quality of text representation normally dominates the model performance (Pennington et al., 2014; Song et al., 2017, 2018; Peters et al., 2018; Song and Shi, 2018; Devlin et al., 2019), in our experiments, we test two types of widely used and powerful encoders, i.e., BiLSTM and Transformer (Vaswani et al., 2017), for $\mathcal{SE}$ and $\mathcal{PE}$. For the embeddings, we use AraBERT (Antoun et al., 2020) and the large version of ZEN 2.0 (Song et al., 2021) with their default settings (i.e. 12 layers of multi-head attentions with 768 dimensional hidden vectors for AraBERT and 24 layers of multi-head attentions with 1024 dimensional hidden vectors for ZEN 2.0) to perform the initialization (we use the output of the last layer).[6] We train our model for 20 epochs in total, with the first 10 for the first training stage and the rest for the second stage. Particularly, in the second training stage, we evaluate our model on the development set for every 100 steps to locate the best performing model. For evaluation, we follow previous studies (Abandah et al., 2015b; Fadel et al., 2019b) to use diacritization error rate (DER) and word error rate (WER) with and without considering the case ending.[7] We also use diacritization accuracy following Zalmout and Habash (2017, 2019, 2020).[8]

3.2 Overall Results

In the main experiment, we run the baselines and our models using different configurations (i.e., using AraBERT or ZEN 2.0 embeddings and using BiLSTM or Transformer encoders) with and without $\mathcal{RD}$ and $\mathcal{AT}$. The experimental results (DER and WER with and without considering the case endings, and accuracy) on the test sets of ATB and Tashkeela are reported in Table 1.[9]

There are several observations. First, under different configurations (i.e., using AraBERT or ZEN 2.0 and with BiLSTM or Transformer encoders), $\mathcal{RD}$ improves the baseline on both datasets, which shows that $\mathcal{RD}$ is effective to help diacritization with auto-generated knowledge even if they follow different criterion. Second, further consistent improvement can be observed when $\mathcal{AT}$ is applied on top of $\mathcal{RD}$, with only 3K (0.015‰ of the entire model size) more trainable parameters required to achieve this effect.[10] These observations confirm the effectiveness of forcing $\mathcal{SE}$ to learn from the information shared by gold and auto-generated labels with an appropriate model design.

[4] We illustrate the dataset details in Appendix B.

[5] http://qatsdemo.cloudapp.net/farasa/

[6] We obtain the pre-trained official AraBERT model from https://github.com/aub-mind/arabert and the Arabic version of ZEN 2.0 (large) from https://github.com/sinovation/ZEN2.

[7] We show details of DER and WER in Appendix C.

[8] We report the hyper-parameter settings of different models and the best combinations of them in Appendix D.

[9] Their dev set's results and the mean and standard deviation of test set results are reported in Appendix E and F.

[10] Model sizes are reported in Appendix G.

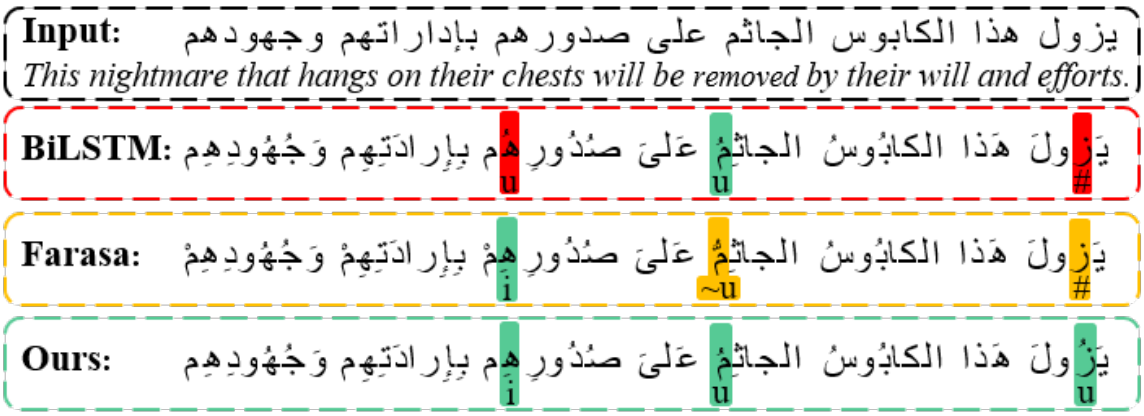


Figure 2: An example input sentence and its diacritization results ("$\sim u$", "#", "i" and "u") from Farasa, BiLSTM, and our approach (BiLSTM+$\mathcal{RD}$+$\mathcal{AT}$) with AraBERT. All results matching gold labels are highlighted in green; the mismatching results from Farasa and BiLSTM are in orange and red, respectively.

In addition, we also compare the results of our best models (with $\mathcal{RD}$ and $\mathcal{AT}$) with previous studies (including Farasa's results) on the test sets of both datasets. The results are shown in Table 2, where our model with BiLSTM encoder outperforms previous models and achieves state-of-the-art performance on both datasets.

3.3 Case Study

To explore how our approach with $\mathcal{RD}$ and $\mathcal{AT}$ leverage auto-generated knowledge, we conduct a case study on an example sentence from the test set of ATB. The input and its diacritization results from Farasa, the BiLSTM baseline, and our approach with AraBERT (BiLSTM+$\mathcal{RD}$+$\mathcal{AT}$) are illustrated in Figure 2, where the correct diacritization results are highlighted in green, and the incorrect ones from Farasa and BiLSTM are highlighted in orange and red, respectively. It is clearly observed that our approach leverages the necessary information learned from Farasa (i.e., the "$\sim u$" label) and prevents its unreliable results from affecting the final diacritics. Specifically, for the highlighted Arabic character "ه", where the Farasa output suggests the diacritic "i" (*kasra*), our approach leverages this knowledge and corrects the BiLSTM baseline. For the other two highlighted characters, although the Farasa output (i.e., "$\sim u$" (*Shadda+Damma*) for "م" and "#" (*No Diacritic*) for "ز") also produces diacritization results that are different from the BiLSTM baseline and do not match the gold standard, our approach is able to learn from their patterns and make correct predictions. Therefore, although Farasas output does not match the gold labels in most cases (see the Farasa results in Table 2), the proposed $\mathcal{RD}$ and $\mathcal{AT}$ can leverage such knowledge and improve the main tagger accordingly.

4 Conclusion

In this paper, we propose to incorporate auto-generated knowledge (diacritization labels in another criterion) for Arabic diacritization with regularized decoding and adversarial training. In detail, the regularized decoding treats the auto-generated knowledge as separate "gold" labels and learns to predict them in another decoding process; the adversarial training is used to ensure that the shared information from gold and auto-generated labels are learned to help diacritization. With the regularized decoding and adversarial training, the main tagger in our approach is able to smartly leverage auto-generated knowledge provided by an existing diacritization tagger. Experimental results on two benchmark datasets illustrate the validity and effectiveness of our model, where state-of-the-art performance is obtained on both datasets.

Acknowledgements

This work is supported by Shenzhen Institute of Artificial Intelligence and Robotics for Society under the project "Automatic Knowledge Enhanced Natural Language Understanding and Its Applications" (AC01202101001).

References

Gheith Abandah and Asma Abdel-Karim. 2019. Accurate and Fast Recurrent Neural Network Solution for the Automatic Diacritization of Arabic Text. *Jordanian Journal of Computers and Information Technology*, 6.

Gheith A Abandah, Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al-Taee. 2015a. Automatic Diacritization of Arabic Text using Recurrent Neural Networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Gheith A. Abandah, Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad T. Jamour, and Majid A. Al-Taee. 2015b. Automatic diacritization of Arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18:183–197.

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A Fast and Furious Segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2019. Efficient convolutional neural networks for diacritic restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1442–1448.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2020. A multitask learning approach for diacritic restoration. *arXiv preprint arXiv:2006.04016*.

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. *ArXiv*, abs/2003.00104.

Alaa Khaled Radwan Arabiyat. 2015. Automatic Arabic Text Diacritization Using Recurrent Neural Networks. Master's thesis, The University of Jordan.

Yonatan Belinkov and James Glass. 2015. Arabic Diacritization with Recurrent Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285.

Guimin Chen, Yuanhe Tian, and Yan Song. 2020. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279.

Guimin Chen, Yuanhe Tian, Yan Song, and Xiang Wan. 2021. Relation Extraction with Type-aware Map Memories of Word Dependencies. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Mohamed Eldesouki. 2020. Arabic Diacritic Recovery Using a Feature-Rich bilstm Model. *ArXiv*, abs/2002.01207.

Kareem Darwish, Hamdy Mubarak, and Ahmed Abdelali. 2017. Arabic Diacritization: Stats, Rules, and Hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. Ldc arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Drago, Burileanu, Vladimir Popescu, Cristian Negrescu, and Aurelian Dervi. 2008. Automatic Diacritic Restoration for a Tts-based E-mail Reader Application.

Ali Fadel, Ibraheem Tuffaha, Mahmoud Al-Ayyoub, et al. 2019a. Arabic Text Diacritization Using Deep Neural Networks. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–7.

Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019b. Neural Arabic Text Diacritization: State of the Art Results and a Novel Approach for Machine Translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 215–225.

Nizar Habash and Owen Rambow. 2007. Arabic Diacritization through Full Morphological Tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56, Rochester, New York.

Ehab Hermena, Denis Drieghe, Sam Hellmuth, and Simon Liversedge. 2015. Processing of Arabic Diacritical Marks: Phonological-Syntactic Disambiguation of Homographic Verbs and Visual Crowding Effects. *Journal of experimental psychology. Human perception and performance*, 41.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank : Building a Large-Scale Annotated Arabic Corpus.

Angrosh Mandya, Danushka Bollegala, and Frans Coenen. 2020. Graph Convolution over Multiple Dependency Sub-graphs for Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6424–6435.

Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. 2020. Improving Named Entity Recognition with Attentive Ensemble of Syntactic Information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana.

Anas Shahrour, Salam Khalifa, and Nizar Habash. 2015. Improving Arabic Diacritization through Syntactic Analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1309–1315.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song and Shuming Shi. 2018. Complementary Learning of Word Embeddings. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4368–4374.

Yan Song, Shuming Shi, and Jing Li. 2018. Joint Learning Embeddings for Chinese Words and Their Components via Ladder Structured Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4375–4381.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588.

Yuanhe Tian, Guimin Chen, and Yan Song. 2021a. Aspect-based Sentiment Analysis with Type-aware Graph Convolutional Networks and Layer Ensemble. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2910–2922, Online.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021b. Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Yuanhe Tian, Wang Shen, Yan Song, Fei Xia, Min He, and Kenli Li. 2020a. Improving Biomedical Named Entity Recognition with Syntactic Information. *BMC Bioinformatics*, 21:1471–2105.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Joint Chinese Word Segmentation and Part-of-speech Tagging via Multi-channel Attention of Character N-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Nasser Zalmout and Nizar Habash. 2017. Don't throw Those Morphological Analyzers Away Just Yet: Neural Morphological Disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 704–713, Copenhagen, Denmark.

Nasser Zalmout and Nizar Habash. 2019. Adversarial multitask learning for joint multi-feature and multi-dialect morphological modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1775–1786, Florence, Italy.

Nasser Zalmout and Nizar Habash. 2020. Joint Diacritization, Lemmatization, Normalization, and Fine-Grained Morphological Tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. *Data in Brief*, 11:147 – 151.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584.

Appendix A. Diactritization Labels

Table 3 presents the 15 diacritization labels used in our study, following Fadel et al. (2019b).

Class Label	Class Name	Class Shape	Class Label	Class Name	Class Shape
#	No diacritics	بـ	~	Shadda	بّ
a	Fatha	بَ	~a	Shadda+Fatha	بَّ
i	Kasra	بِ	~i	Shadda+Kasra	بِّ
o	Sukun	بْ	~u	Shadda+Damma	بُّ
u	Damma	بُ	~K	Shadda+Kasratan	بٍّ
K	Kasratan	بٍ	~F	Shadda+Fathatan	بًّ
F	Fathatan	بً	~N	Shadda+Dammatan	بٌّ
N	Dammatan	بٌ			

Table 3: Diactritization labels used in this study.

Appendix B. Datasets

In our experiments, We use two benchmark datasets, i.e., ATB (Arabic Treebank Part 1, 2, and 3)[11] (Maamouri et al., 2004) and Tashkeela[12] (Zerrouki and Balla, 2017). For ATB, we follow the same data split policy as Diab et al. (2013), which is based on the 10-80-10 rule. That is, we firstly split each part of ATB into three portions (with each portion containing 10%, 80%, and 10% of documents, respectively). Then, we combine the first, second, and third portion of all three parts to form the development, training, and test set, respectively. For Tashkeela, we use the cleaned version[13] from Fadel et al. (2019b) with the standard train/dev/test split. The statistics of the datasets in terms of the number of words, lines, and the average number of characters in each word are reported in Table 4.

	ATB			Tashkeela		
	Train	Dev	Test	Train	Dev	Test
Word #	503K	63K	63K	2.1M	102K	107K
Line #	15.7K	1.9K	1.9K	50K	2.5K	2.5K
C/W	4.37	4.31	4.35	3.97	3.97	3.97

Table 4: Statistics of the benchmark datasets, where the number of words and lines, and average characters per word (C/W) are reported.

[11] We download the ATB part 1, 2 and 3 are from https://catalog.ldc.upenn.edu/LDC2010T13, https://catalog.ldc.upenn.edu/LDC2011T09 and https://catalog.ldc.upenn.edu/LDC2010T08.

[12] https://github.com/AliOsm/arabic-text-diacritization/tree/master/dataset

[13] We download the data from https://github.com/AliOsm/arabic-text-diacritization/tree/master/dataset.

Appendix C. Evaluation of DER and WER

It is worth noting that previous studies (Zitouni et al., 2006; Arabiyat, 2015; Fadel et al., 2019a; Abandah and Abdel-Karim, 2019; Alqahtani et al., 2019, 2020) use different methods to compute diacritic error rate (DER) for ATB and Tashikeela datasets. Therefore, we follow the schema in Zitouni et al. (2006); Arabiyat (2015); Abandah and Abdel-Karim (2019) to compute DER for ATB and follow Fadel et al. (2019a); Alqahtani et al. (2019, 2020) to compute that for Tashikeela.

Specifically, for ATB, we compute DER by: (1) all words are counted including numbers and punctuators; (2) each latter or digit in a word is a potential host for a set of diacritics; and (3) all diacritics on a single letter are counted as a single binary (True or False) choice. For Tashikeela, the schema is similar to the one for ATB but all non-Arabic letters are ignored in computing DER because they do not hold a diacritic. For word error rate (WER), the way to compute it is identical for both datasets, where the diacritization result for an Arabic word is regarded as incorrect if there is s at least one incorrectly restored diacritic. We follow previous studies (Abandah et al., 2015b; Fadel et al., 2019a) to evaluate our results in terms of diacritic error rate (DER) and word error rate (WER). We use the implementation[14] provided by Fadel et al. (2019a) to compute DER (with two criteria) and WER of different models on both datasets, where the DER and WER with and without considering the case endings are both included in our evaluation.

Appendix D. Hyper-parameter Settings

Table 5 reports the hyper-parameters tested in training our models. We test all combinations of them for each model and use the one achieving the highest F1 score in our final experiments.

Hyper-parameters	Values
Learning Rate	$1e-5, \mathbf{3e-5}, 5e-5$
Warmup Rate	**0.06**
Dropout Rate	**0.1**
Batch Size	$16, \mathbf{32}, 64$

Table 5: The hyper-parameters tested in tuning our models. The best ones used in our final experiments are highlighted in boldface.

[14] https://github.com/AliOsm/arabic-text-diacritization/blob/master/helpers/diacritization_stat.py.

Appendix E. Experimental Results on the Development Set

Table 6 reports the DER and WER (with case ending) of different models evaluated on the development set of ATB and Tashkeela.

	ATB		Tashkeela	
	DER	WER	DER	WER
BiLSTM	2.52	7.00	2.58	7.66
+$\mathcal{RD}$	2.46	6.47	2.20	6.47
+$\mathcal{RD}$+$\mathcal{AT}$	2.14	5.65	2.12	6.30
Transformer	2.46	6.79	2.71	7.96
+$\mathcal{RD}$	2.35	6.28	2.07	6.08
+$\mathcal{RD}$+$\mathcal{AT}$	2.09	5.50	2.05	6.03

(a) AraBERT

	ATB		Tashkeela	
	DER	WER	DER	WER
BiLSTM	2.41	6.67	2.51	7.45
+$\mathcal{RD}$	2.21	6.00	2.09	6.16
+$\mathcal{RD}$+$\mathcal{AT}$	2.03	5.43	2.29	6.22
Transformer	2.39	6.49	2.65	7.79
+$\mathcal{RD}$	2.03	5.40	2.21	6.09
+$\mathcal{RD}$+$\mathcal{AT}$	1.96	5.24	2.01	5.87

(b) ZEN 2.0

Table 6: DER and WER (with case ending) of models with different configurations (i.e., based on BiLSTM and Transformer) evaluated on the development set of ATB and Tashkeela.

Appendix F. Mean and Deviation of the Results

In the experiments, we test models with different configurations. For each model, we train it with the best hyper-parameter setting using five different random seeds. We report the mean (μ) and standard deviation (σ) of DER and WER (with case ending) on the test set of ATB and Tashkeela in Table 7.

Appendix G. Model Size and Running Speed

Table 8 reports the number of trainable parameters and the inference speed (lines per second) of the baseline (i.e., BiLSTM and Transformer encoder with and without regularized decoding ($\mathcal{RD}$)) and our models with both $\mathcal{RD}$ and adversarial training ($\mathcal{AT}$) on ATB and Tashkeela. All models are performed on NVIDIA Quadro RTX 6000 GPUs.

Models	ATB				Tashkeela			
	DER		WER		DER		WER	
	μ	σ	μ	σ	μ	σ	μ	σ
BiLSTM	2.33	0.0115	6.70	0.4082	2.61	0.0004	7.67	0.0020
+$\mathcal{RD}$	2.14	0.0183	5.92	0.1669	2.30	0.0114	6.79	0.1022
+$\mathcal{RD}$+$\mathcal{AT}$	1.94	0.0053	5.38	0.0589	2.21	0.0080	6.40	0.0683
Transformer	2.27	0.0413	6.58	0.5590	2.84	0.0537	8.15	0.3696
+$\mathcal{RD}$	2.10	0.0008	5.92	0.0122	2.14	0.0003	5.96	0.1934
+$\mathcal{RD}$+$\mathcal{AT}$	1.88	0.0026	5.29	0.0651	2.12	0.0024	6.20	0.0323

(a) AraBERT

Models	ATB				Tashkeela			
	DER		WER		DER		WER	
	μ	σ	μ	σ	μ	σ	μ	σ
BiLSTM	2.19	0.0374	6.23	0.0613	2.53	0.0411	7.42	0.0736
+$\mathcal{RD}$	1.99	0.0170	5.66	0.0450	2.12	0.0419	6.04	0.0510
+$\mathcal{RD}$+$\mathcal{AT}$	1.85	0.0327	5.15	0.0988	2.08	0.0408	5.95	0.0665
Transformer	2.08	0.0249	5.90	0.0860	2.69	0.0205	7.70	0.0411
+$\mathcal{RD}$	1.87	0.0205	5.15	0.0327	2.02	0.0531	5.78	0.1307
+$\mathcal{RD}$+$\mathcal{AT}$	1.80	0.0249	4.96	0.0655	1.93	0.0490	5.65	0.0829

(b) ZEN 2.0

Table 7: The mean μ and standard deviation σ of DER and WER (with case ending) of all models (i.e., based on BiLSTM or Transformer with $\mathcal{RD}$ and $\mathcal{AT}$) on the test set of ATB and Tashkeela for Arabic diacritization.

	ATB		Tashkeela	
	Para.	Speed	Para.	Speed
BiLSTM	158,840K	55.6	158,840K	54.3
+$\mathcal{RD}$	206,162K	36.4	206,162K	28.5
+$\mathcal{RD}$+$\mathcal{AT}$	206,165K	25.6	206,165K	22.9
Transformer	146,235K	70.7	146,235K	71.1
+$\mathcal{RD}$	168,335K	37.7	168,335K	34.3
+$\mathcal{RD}$+$\mathcal{AT}$	168,337K	29.0	168,337K	27.2

(a) AraBERT

	ATB		Tashkeela	
	Para.	Speed	Para.	Speed
BiLSTM	839,026K	30.2	839,026K	29.8
+$\mathcal{RD}$	872,730K	20.4	872,730K	19.7
+$\mathcal{RD}$+$\mathcal{AT}$	872,734K	14.8	872,734K	13.3
Transformer	830,612K	39.6	830,612K	37.2
+$\mathcal{RD}$	847,471K	25.8	847,471K	24.0
+$\mathcal{RD}$+$\mathcal{AT}$	847,473K	21.1	847,473K	19.4

(b) ZEN 2.0

Table 8: Numbers of trainable parameters (Para.) in different models and the inference speed (sentences per second) of these models on the test sets of both datasets. $\mathcal{RD}$ and $\mathcal{AT}$ represent the proposed regularized decoding and adversarial training, respectively.

When is Char Better Than Subword: A Systematic Study of Segmentation Algorithms for Neural Machine Translation

Jiahuan Li * **Yutong Shen** * **Shujian Huang** † **Xinyu Dai** **Jiajun Chen**
National Key Laboratory for Novel Software Technology, Nanjing University, China
{lijh,shenyt}@smail.nju.edu.cn,
{huangsj,daixinyu,chenjj}@nju.edu.cn

Abstract

Subword segmentation algorithms have been a *de facto* choice when building neural machine translation systems. However, most of them need to learn a segmentation model based on some heuristics, which may produce suboptimal segmentation. This can be problematic in some scenarios when the target language has rich morphological changes or there is not enough data for learning compact composition rules. Translating at fully character level has the potential to alleviate the issue, but empirical performances of character-based models has not been fully explored. In this paper, we present an in-depth comparison between character-based and subword-based NMT systems under three settings: translating to typologically diverse languages, training with low resource, and adapting to unseen domains. Experimental results show strong competitiveness of character-based models. Further analyses show that compared to subword-based models, character-based models are better at handling morphological phenomena, generating rare and unknown words, and more suitable for transferring to unseen domains.

1 Introduction

Neural machine translation (NMT) has achieved great success in recent years. Modern NMT systems typically operate on subword level, using segmentation algorithms such as *byte pair encoding* (BPE) (Sennrich et al., 2016) or Morfessor (Creutz and Lagus, 2002). Compared to word-level models, subword segmentation helps overcome the out-of-vocabulary (OOV) problem and make better use of morphological information in the surface form.

Despite their empirical effectiveness, subword algorithms may produce improper segmentation due to their data-dependent nature. NMT models are typically robust to such errors when trained on large corpora or the target language is regular in morphological changes, like French or German. However, the problem will arise when such conditions are not met, i.e. there is not enough data for learning compact composition rules or the target language is morphologically rich and complex.

An alternative segmentation choice is to use fully character-level (CHAR) models (Lee et al., 2017; Cherry et al., 2018; Gupta et al., 2019; Gao et al., 2020; Banar et al., 2020), which has the potential to alleviate above issues. CHAR does not need to *learn* any segmentation rules and keeps all available information in the surface form, avoiding the risk of information loss due to improper segmentation. What is more, the main pain point of CHAR that it takes too long to train is less obvious in above settings since there is not as much data as in the rich resource setting. However, there has not been a comprehensive study in these settings.

In this paper, we conduct a systematic comparison between CHAR and other subword algorithms, e.g. BPE and Morfessor. Experiments show strong competitiveness of CHAR under three settings: translating to typologically diverse languages (Section 2), training with low resource (Section 3), and adapting to distant domains (Section 4). Further analyses show that compared to subword algorithms, the benefits of CHAR mainly come from better capture of the morphological phenomena, better generation of rare and unknown words, and better translation of domain-specific words.

2 Translation Across Typologically Diverse Languages

Human languages are known to exhibit diverse morphological phenomena, which could serve as a principle to classify languages into different morphological categories, such as *fusional*, *agglutinative*, *introflexive* and *isolating*. While previous

* Equal contribution
† Corresponding author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 543–549
August 1–6, 2021. ©2021 Association for Computational Linguistics

		Word	Char	BPE	Morf.
F.	Fr	39.1/.580	40.1/.589	**41.2/.597**	39.6/.592
	Ro	31.1/.487	**33.9/.526**	32.9/.517	30.6/.517
A.	Fi	21.9/.412	**23.5/.487**	22.3/.472	21.7/.466
	Tr	19.8/.396	**22.8/.456**	21.1/.440	16.9/.437
In.	Hi	14.0/.262	**15.6/.290**	14.8/.285	14.8/.276
	Ar	22.5/.451	**24.7/.491**	23.9/.481	23.5/.481
Is.	Vi	21.6/.374	**22.5/.385**	22.2/.381	21.1/.373
	Ml	22.9/.324	**25.0**/.349	24.3/.347	24.1/**.356**

Table 1: BLEU/chrF3 scores of systems translating from English to languages of different morphological categories, using different segmentation algorithms. Best score in each line is shown in bold.

	Word	Char	BPE	Morf.
Comp. adj.	55.6	**70.8**	63.0	60.0
Det. poss.	49.6	**83.0**	78.0	78.4
Pron. hum	60.6	**67.0**	66.2	66.6
Local case	36.6	**61.8**	50.6	47.6
Pron. gender	73.6	76.6	**79.0**	**79.0**
Verb neg	96.6	97.2	**98.4**	98.0
Preposition	33.8	**69.2**	60.2	64.2
Future tense	51.4	43.8	**53.8**	50.8
Past tense	83.2	**91.8**	87.4	90.8
Pron. plural	74.6	**79.2**	77.4	75.2
Noun plural	48.8	**76.0**	62.8	60.8
Det. definite	38.4	38.8	40.8	**44.8**
Named Ent.	9.2	**70.4**	66.4	30.2
Number	65.4	**96.6**	91.2	77.8

Table 2: Performance of different segmentation algorithms on the MorphEval En-Fi benchmark. Each row represents a kind of morphological phenomenon.

works only focus on performances of character-level models when translating to fusional and agglutinative languages (Gupta et al., 2019; Libovický and Fraser, 2020), we conduct a comprehensive study covering all four morphological categories.

2.1 Experiment Setup

Dataset We consider the translation from English to eight target languages representing four morphological categories, i.e. French (**Fr**) and Romanian (**Ro**) for *fusional*, Finnish (**Fi**) and Turkish (**Tr**) for *agglutinative*, Hebrew (**He**) and Arabic (**Ar**) for *introflexive*, and Vietnamese (**Vi**) and Malaysian (**Ml**) for *isolating*. We use OPUS-100 corpus[1] (Tiedemann, 2012), which consists of 1M parallel sentences for each language pair.

Model and Hyperparameters We use the Transformer architecture (Vaswani et al., 2017) throughout all experiments. To ensure results' reliability , we run an exhaustive search of hyperparameters including batch size and learning rate. Detailed hyperparameters can be found in Appendix A.

2.2 Results

The results are listed in Table 1. We can see that CHAR outperforms other algorithms in 7 out of 8 languages in terms of BLEU (Papineni et al., 2002) and chrF3 (Popović, 2015), showing strong competitiveness of CHAR's ability across languages. The only exception is the En-Fr language pair, which are known to be quite similar and is beneficial for BPE to learn a joint segmentation model.

It is intuitive that BPE and Morfessor cannot outperform CHAR on introflexive languages (Hi, Ar). Introflexive languages follows non-concatenative morphology (McCarthy, 1981), i.e. grammatical information is conveyed by directly modifying the root words. This makes it hard for linear segmenting methods such as BPE and Morfessor to work well. This finding is also consistent with previous research on other tasks (Zhu et al., 2019).

For isolating languages (Vi, Ml), there are rare morphological phenomena indicating grammatical relations, so segmentation algorithms do not greatly affect the performance. We can see that the two open-vocabulary segmentation algorithms (CHAR, BPE) show comparable performances.

Surprisingly, even for highly agglutinative languages such as Finnish and Turkish, which has very regular morphological changes by adding affixes or suffixes, CHAR still achieves better performance.

2.3 Analysis on MorphEval

To understand where the advantages of CHAR model come from, we take Finnish as an example and evaluate the morphological competence of different models using MorphEval test suites (Burlot et al., 2018). MorphEval generates pairs of source sentences that differ by one kind of morphological phenomena, and assesses a MT system's ability by computing the percentage of its generated target sentences that convey as the source sentences. Higher accuracy means the model is more sensitive to the current morphological phenomenon.

As shown in Table 2, CHAR performs the best in 10 out of 14 tests. Among these 10 tests, in comparative adjectives, possessive determiner, local postposition case, preposition case, plural nouns, CHAR surpasses other algorithms notably by at least 5% accuracy. This indicates CHAR's strong ability to capture the fine-grained morphological

[1] http://data.statmt.org/opus-100-corpus/v1.0/supervised/

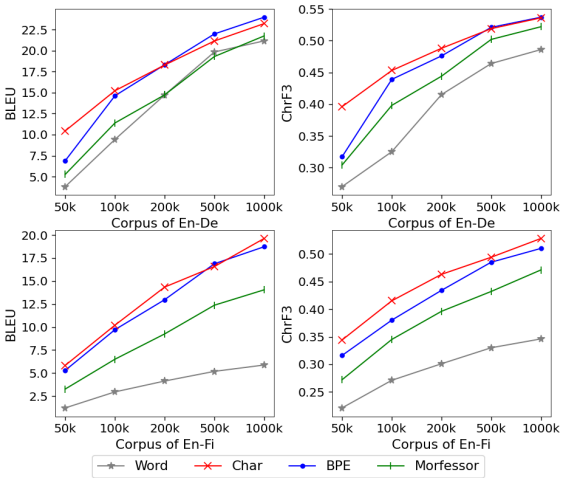


Figure 1: BLEU and chrF3 score curves using different amount of parallel data for training En-De (above) and En-Fi (below) translation systems.

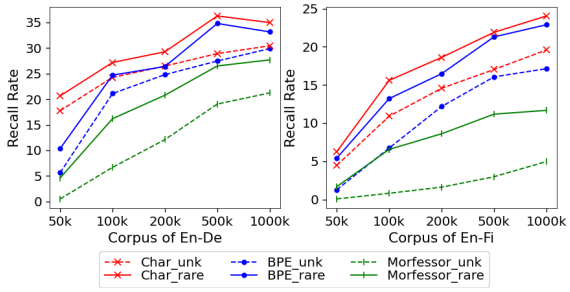


Figure 2: Recall rates of unknown and rare words generated by systems based on different tokenizers models. Words appearing no more than 5 times in the training set are considered as rare words.

phenomena, which is crucial for MT models when translating into morphologically rich languages.

Interestingly, three of four morphological phenomena on which CHAR falls behind are so-called *stability* features (Burlot et al., 2018), which are expressed differently in the source language but should be expressed identically in the target language[2]. The disadvantage of CHAR in this kind of phenomena shows CHAR-based model may be less robust to lexical changes to source-side changes, and the reason needs to be further researched.

3 Translation with Low Resource

Subword algorithms help alleviate the OOV problem. However, most of them are based on heuristics and may produce wrong segmentation. While this problem is not so evident when there is enough data to learn robust composition rules, in low-resource setting it could be a different story and their effectiveness should be examined. While for CHAR, pure character sequences can directly provide all the information to the model for learning the composition rules. Therefore a prudent choice of segmentation should be studied in this setting.

3.1 Experiment Setup

We perform evaluation on WMT14 En-De[3] and WMT17 En-Fi[4] dataset. Datasets of size 50k, 100k, 200k, 500k and 1000k are subsampled from the original training dataset and serve as training data of different resource conditions. For validation and test, we use the original development and test split.

Previous works (Sennrich and Zhang, 2019; Nguyen and Chiang, 2017) show that in low resource settings the evaluation results can be sensitive to model size (e.g. hidden dimension, layer number) and the number of BPE merges k, so we run an additional search of hidden dimension, layer number and k, and report the best results in this section. See Appendix A for details.

3.2 Results

We evaluate models with BLEU and chrF3. The results are showed in Figure 1. In general, the performances of CHAR and BPE are on par, and are better than Word and Morfessor. In different data conditions, the results varies.

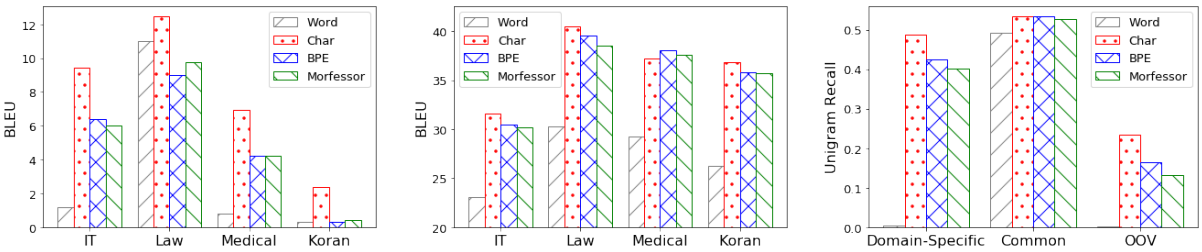
medium-resource When there are plenty resources, e.g. 500k and 1000k, the performance of CHAR and BPE are comparable but different for different language pairs. For En-Fi, CHAR is better than BPE. It is because morphological changes in Finnish are quite complex. More fine-grained segmentation like CHAR is needed to learn corresponding rules. Conversely, German's morphological changes are so regular that BPE can learn most of merging rules, making them performing better.

low-resource When the corpus size is 50k to 200k, CHAR performs the best among four segmentation methods. BPE and Morfessor usually regard frequently occurring words as single tokens, many of which contain rich morphological information. This, together with the improper segmentation problem, prevents NMT models from learning correct composition rules, damaging the model's generalization ability on rare and unknown words. In low resource setting this problem would be more se-

[2]For example, English uses *he/she* to convey the masculine/feminine contrast, but Finnish uses the same pronoun *hän* regardless of the gender of the antecedent.

[3]http://www.statmt.org/wmt14/translation-task.html

[4]http://statmt.org/wmt17/translation-task.html



(a) Average OOD BLEU (No Adapt). (b) Average OOD BLEU (Finetune). (c) Recall of different word types.

Figure 3: Domain robustness of translation systems based on different segmentation algorithms.

vere, since there are much more rare and unknown words but not enough data for learning compact composition rules.

Compared with subwords, character-based models learn combinations directly from character sequences. Not limited to fixed char sequence patterns in subwords, more words with different morphological changes can be generated through CHAR. Therefore, CHAR can learn more correct composition rules than subword-based model, leading to better translation of rare and unknown words.

3.3 Analysis on Rare and Unknown Words

To further support the above analysis, we evaluate the translation quality of rare and unknown words by calculating their recall rates. The results are showed in Figure 2. We can see that CHAR has achieved the highest recall rates of rare and unknown words. Although, as the resource increases, the gap between CHAR and BPE is shrinking gradually, the results can still prove that CHAR can capture more morpheme information, performing better at generating rare and unknown words.

4 Translation Across Distant Domains

Domain robustness (Müller et al., 2020), which refers to models' generalization ability on unseen domains, is important for NMT applications. However, subword algorithms need to learn segmentation rules from a given corpus, which may be domain-specific. When applied to a new domain, they may improperly segment target-domain specific words, hurting the domain robustness. In contrast, CHAR does not suffer from the issue. In this section, we investigate how different segmentation algorithms affect NMT models' domain robustness.

4.1 Experiment Setup

We use the same corpora as (Koehn and Knowles, 2017), which is a De-En dataset covering subsets of four domains: *Law*, *Medical*, *IT* and *Koran*. Following Koehn and Knowles (2017), each time we train a source domain model on one of four subsets and report results on test sets of the other three domain. We experiment in two settings: **No Adapt** and **Finetune**. The first one involves no target domain data, while the latter uses randomly sampled 100k sentence pairs from target domain data to finetune the source domain model.

4.2 Results

We report the average out-of-domain (OOD) BLEU scores of NMT systems based on different segmentation algorithms in Figure 3a and Figure 3b. As can be seen from the figure, CHAR surpasses other algorithms in almost all settings, except when finetuning from Medical to others. This illustrates the suitability of CHAR for domain robustness, especially when there is no enough data for adaptation.

4.3 Analysis on Different Types of Words

To understand the advantages of CHAR, we take the setting of finetuning from *IT* to *Medical* as an example and analyze performances on different types of words. Specifically, we divide words in the test set into three types: **(1) Domain-specific** words occur only in the target domain training data; **(2) Common** words occur in both the source and target domain training data; **(3) OOV** words *do not* occur in both training data.

The result can be seen in Figure 3c. CHAR achieves better performance on OOV words, which is consistent with findings in Section 3. While performances of CHAR and subword-based algorithms are on par on common words, CHAR outperforms the others by a large margin on domain-specific words. This suggests that the advantage of CHAR mainly comes from the correct translation of domain-specific and OOV words, which may be segmented improperly by subword algorithms.

	Word	Char	BPE	Morf.	BPE-D
No adapting	11.03	**12.46**	9.02	9.74	11.11
Finetune	30.26	**40.53**	39.53	38.49	40.26

Table 3: Average OOD BLEU of models based on different subword algorithms when adapting from *Law* to other domains. BPE-D: BPE-dropout (Provilkov et al., 2020)

4.4 Comparison with Advanced Segmentation Algorithms

Although we focus on deterministic segmentation algorithms in this paper, there are more advanced ones such as BPE-dropout (Provilkov et al., 2020) and subword regularization (Kudo, 2018), which produce multiple segmentation candidates when training and show improved performance. Therefore, we also conduct experiments comparing CHAR with BPE-dropout in terms of domain adaptation performance. We take the setting of adapting from *Law* to other domains and report results in Table 3. As can be seen, although BPE-dropout surpasses BPE by a large margin, CHAR still achieves the best performance, which again shows the superiority of CHAR.

5 Related Work

Character-level neural machine translation has received growing attention in recent years. Lee et al. (2017) first propose a fully character-level NMT model based on recurrent encoder-decoder architecture and convolutional layers, which shows a promising results. Gao et al. (2020) propose to incorporate convolution layers in the more advanced Transformer architecture and show their model can learn more robust character-level alignments.

However, translating at character level may incur significant computational overhead. Therefore, later works on character-level NMT (Cherry et al., 2018; Banar et al., 2020) mainly focus on reducing computation cost of them. Cherry et al. (2018) show that by employing source sequence compression techniques, the quality and efficiency of character-based models can be properly balanced. Banar et al. (2020) share the same idea as Cherry et al. (2018) but build their models using Transformer architecture. Our work differs from theirs in that we aim to analyze the performance of existing models instead of exploring novel architectures.

There are also several researches on comparison between CHAR and other subword algorithms (Durrani et al., 2019; Gupta et al., 2019). Durrani et al. (2019) compare character-based models and subword-based models in terms of representation quality, and find that representation learned by the former are more suitable for modeling morphology, and more robust to noisy input. Gupta et al. (2019) investigate the performance of different segmentation algorithms when using Transformer architecture, and find that character-based models can achieve better performance when translating noisy text or text from a different domain. Our finds are consistent with them, yet we conduct a more large-scale and in-depth analysis by covering language pairs from more language families and explaining where the advantage of character-based models comes from.

6 Conclusion

We conduct a comprehensive study and show advantages of CHAR over subword algorithms in three settings: translating to typologically diverse languages, translating with low resource, and adapting to distant domains. Note that although we have tried our best to take as much language pairs as possible into consideration, there are certainly a lot of languages remaining uncovered in this paper. However, we believe our experimental results can serve as an evidence of character-based NMT models' strong competitiveness. We hope more attention will be drawn to them, including exploring their more benefits and reducing the possibly higher computation cost in practice.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by the National Key R&D Program of China (No. 2019QY1806), National Science Foundation of China (No. 61772261, U1836221). This work is also partially supported by the research funding from ZTE Corporation.

References

Nikolay Banar, Walter Daelemans, and Mike Kestemont. 2020. Character-level transformer-based neural machine translation. *arXiv preprint arXiv:2005.11239*.

Franck Burlot, Yves Scherrer, Vinit Ravishankar, Ondřej Bojar, Stig-Arne Grönroos, Maarit Koponen, Tommi Nieminen, and François Yvon. 2018. The wmt'18 morpheval test suites for english-czech, english-german, english-finnish and turkish-english.

In *3rd Conference on Machine Translation (WMT 18)*, volume 2, pages 550–564.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Yingqiang Gao, Nikola I. Nikolov, Yuhuang Hu, and Richard H.R. Hahnloser. 2020. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1591–1604, Online. Association for Computational Linguistics.

Rohit Gupta, Laurent Besacier, Marc Dymetman, and Matthias Gallé. 2019. Character-based NMT with transformer. *CoRR*, abs/1911.04997.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Jindřich Libovický and Alexander Fraser. 2020. Towards reasonably-sized character-level transformer NMT by finetuning subword systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2572–2579, Online. Association for Computational Linguistics.

John J McCarthy. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, 12(3):373–418.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Yi Zhu, Ivan Vulić, and Anna Korhonen. 2019. A systematic study of leveraging subword information for learning word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 912–932, Minneapolis, Minnesota. Association for Computational Linguistics.

A Hyperparameters

We conduct a grid search of hyperparameters for the training of Transformer models, including batch size (tokens per batch) and learning rate. For batch size, the searching range is $\{4096, 8192, 16384, 32768\}$. For learning rate, the searching range is $\{5e-5, 1e-4, 5e-4, 1e-3\}$.

Besides, we also experiment with diffrent model size and number of bpe merges k in the low resource settings(50k, 100k, 200k). The searching range of k is $\{2000, 10000\}$. We consider four kinds of model size, i.e. *tiny*, *mini*, *small* and *base*, which differ in their hidden size and transformer layers. The details can be found in Table 4.

	hidden size	layer
tiny	128	2
mini	256	4
small	512	4
base	512	6

Table 4: Detailed hyperparameters for different model sizes.

More than Text: Multi-modal Chinese Word Segmentation

Dong Zhang[1], **Zheng Hu**[2], **Shoushan Li**[1]*, **Hanqian Wu**[2], **Qiaoming Zhu**[1], **Guodong Zhou**[1]
[1]School of Computer Science and Technology, Soochow University, China
[2]School of Computer Science and Engineering, Southeast University, China
{dzhang, lishoushan, qmzhu, gdzhou}@suda.edu.cn
{zhenghu, hanqian}@seu.edu.cn

Abstract

Chinese word segmentation (CWS) is undoubtedly an important basic task in natural language processing. Previous works only focus on the textual modality, but there are often audio and video utterances (such as news broadcast and face-to-face dialogues), where textual, acoustic and visual modalities normally exist. To this end, we attempt to combine the multi-modality (mainly the converted text and actual voice information) to perform CWS. In this paper, we annotate a new dataset for CWS containing text and audio. Moreover, we propose a time-dependent multi-modal interactive model based on Transformer framework to integrate multi-modal information for word sequence labeling. The experimental results on three different training sets show the effectiveness of our approach with fusing text and audio.

1 Introduction

Word segmentation is a fundamental task in Natural Language Processing (NLP) for those languages without word delimiters, e.g., Chinese and many other East Asian languages (Duan and Zhao, 2020). In this paper, we mainly take Chinese language as investigating object, namely CWS. As we know, CWS has been applied as an essential preprocessing step for many other NLP tasks (Zhou et al., 2019; Qiu et al., 2020), such as named entity recognition, sentiment analysis, machine translation, etc.

In the literature, some popular approaches to CWS systems report a high performance at the level of 96%–98%, and these systems typically require a large scale of pre-segmented textual dataset for training. However, the collection of a specific scenario on such large scale is very time-consuming and resource-intensive, such as video monologues and audio broadcast. In these scenarios, there are multiple modalities: text, audio and vision, thus if only using the text seems not a good choice. For example, as shown in Figure 1, if we only read the text"必须不忘初心牢记使命" with no punctuation, it is not easy to make word segmentation immediately. However, if there is the acoustic information, we can observe the obvious stop in spectrum and sonic wave at the middle of "心" and "牢", which provides the facility for CWS.

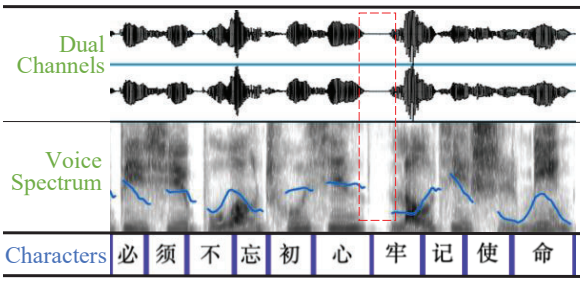


Figure 1: A multi-modal example for CWS. 必须(must) 不(not) 忘(forget) 初(original) 心(heart) 牢记(remember) 使命(mission).

Therefore, in this paper, we propose to performing CWS with multi-modality, namely MCWS, by a time-dependent multi-modal interactive network. Specifically, we first collect a new dataset from an audio and video news broadcast platform and annotate the word boundaries of audio transcription text. Second, we make the text and the audio align as the time stamp of each character, then encode both modalities [1] by Transformer-based framework to capture the intra-modal dynamics. Third, we design a time-dependent multi-modal interaction module for each character step to generate the multi-modal hybrid character representation.

*Corresponding author: lishoushan@suda.edu.cn

[1]Since each video in this platform mainly describe the specific news scene, not the face of the speaker, the visual modality is not useful for word segmentation. Therefore, for the sake of simplicity, we only utilize text and audio to perform CWS.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 550–557
August 1–6, 2021. ©2021 Association for Computational Linguistics

Finally, we leverage the CRF to perform sequence labeling on the basis of the above character representation.

We evaluate our approach on the newly annotated small-scale dataset with different size of training sets. The experimental results demonstrate that our approach performs significantly better than the single-modal state-of-the-art and the multi-modal approaches with early fused features of CWS.

2 Related Work

Xu (2003) first formalize CWS as a sequence labeling task, considering CWS as a supervised learning from annotated corpus with human segmentation. Peng et al. (2004) further adopt standard sequence labeling tool CRFs for CWS modeling, achieving a best performance in their same period. Then, a large amount of approaches based on above settings are proposed for CWS (Li and Sun, 2009; Sun and Xu, 2011; Mansur et al., 2013; Zhang et al., 2013).

Recently, deep neural approaches have been widely proposed to minimize the efforts in feature engineering for CWS (Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015; Cai and Zhao, 2016; Zhou et al., 2017; Yang et al., 2017; Zhang et al., 2017; Ma et al., 2018; Li et al., 2019; Wang et al., 2019a; Fu et al., 2020; Ding et al., 2020; Tian et al., 2020a; Zhao et al., 2020). Among these studies, most of them follow the character-based paradigm to predict segmentation labels for each character in an input sentence. To enhance CWS with neural models, there were studies leverage external information, such as vocabularies from auto-segmented external corpus and weakly labeled data (Wang and Xu, 2017; Higashiyama et al., 2019; Gong et al., 2020).

To our best knowledge, we are first to perform CWS with multi-modality, which can deal with multi-modal scenarios and offers an alternative solution to robustly enhancing neural CWS models.

3 Data Collection and Annotation

We collect the multi-modal data for CWS from a Chinese news reporting platform "Xuexi"[2]. We mainly focus on the audios equipped with machine transcription text. In total, we crawl 120 short videos and segment them into about 2000 sentences. To avoid the contextual influence and augment the robust of designed computing model, we randomly select 250 sentences to annotate the word boundaries, and the remaining data are used to perform semi-supervised or unsupervised learning in the future.

[2]https://www.xuexi.cn/

Items	Size
Sentences	250
Avg. Length (Character)	50.56
Avg. Length (Word)	26.95
Avg. Length (Time)(s)	10.63
Max Length (Character)	382
Max Length (Word)	231
Max Length (Time)(s)	95.06
Total Characters	12640
Total Words	6736
Total Time(s)	2658.16

Table 1: The statistics summary for used data.

We annotate these Chinese audio transcriptions following the CTB word segmentation guidelines by Xia (2000). Two annotators are asked to annotate the data. Due to the clear annotation guideline, the annotation agreement is very high, reaching 98.3%. The disagreement instances are judged by an expert. The statistics of our annotated data are summarized in Table 1.

4 Time-dependent Multi-modal Interactive Network for CWS

In this section, we introduce our proposed multi-modal approach for CWS, namely Time-dependent Multi-modal interactive Network (TMIN), which can capture the interactive semantics between text and audio for better word segmentation. This approach mainly consists of three modules: time-dependent uni-modal interaction, time-dependent multi-modal interaction and CRF labeling. Figure 2 shows the overall architecture of our TMIN.

4.1 Time-dependent Uni-modal Interaction

To better capture the temporal correspondences between different modalities (Zhang et al., 2019; Ju et al., 2020), we first align two modalities by extracting the exact time stamp of each phoneme and character using Montreal Forced Aligner (McAuliffe et al., 2017).

For machines to understand human utterance, they must be first able to understand the intra-modal dynamics (Zadeh et al., 2018; Wang et al., 2019b; Tsai et al., 2019) in each modality, such as the word order and grammar in text, breathe and

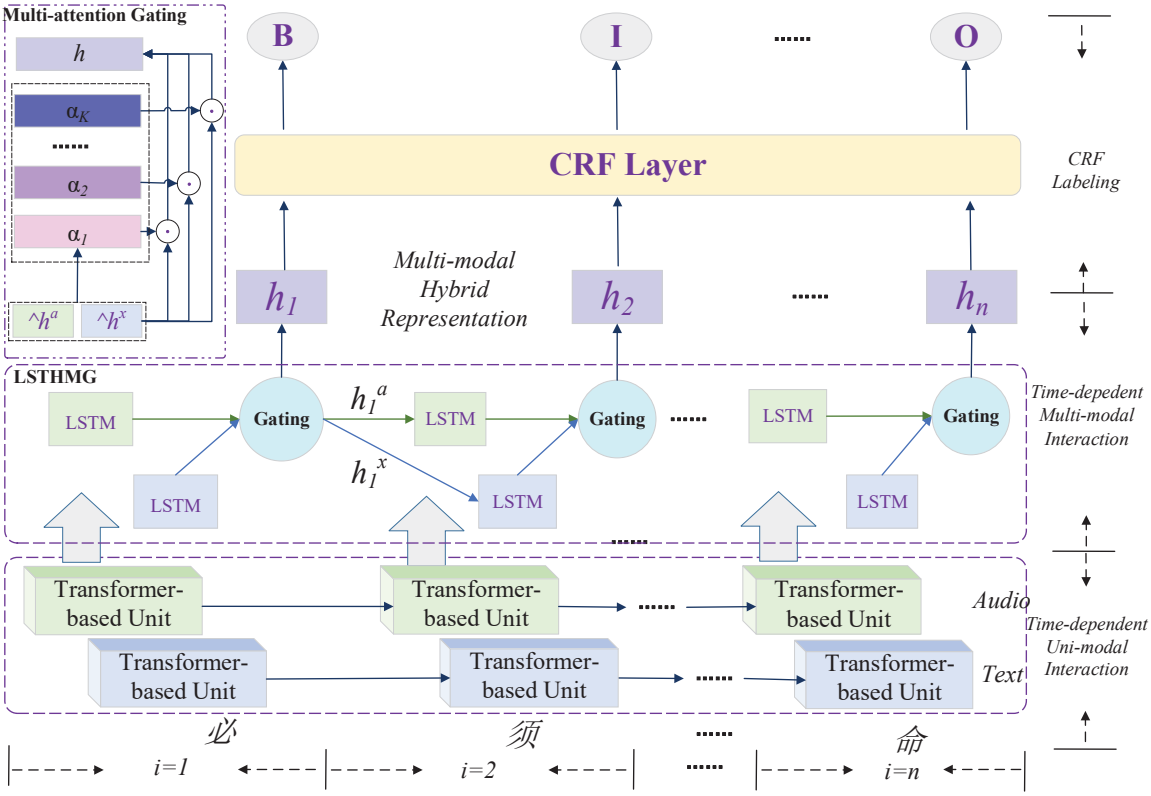


Figure 2: The overview of our proposed TMIN.

tone in audio.

Textual Modality. We use BERT (Devlin et al., 2019) as encoder to perform intra-modal interactions and obtain the contextual character representation. Then, each character of text transcripts can be represented as: $X = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^{n \times d_1}$.

Acoustic Modality. We use a famous audio processing tool, i.e., OpenSMILE (Eyben et al., 2010), to extract the MFCC, LP-coefficients, pure FFT spectrum, etc. from dual channels (Jayram et al., 2002; Sakran et al., 2017), and leverage multiple Transformer layers (Vaswani et al., 2017) to perform intra-modal interactions. Then, each character-level audio feature can be represented as: $A = (a_1, a_2, \cdots, a_n) \in \mathbb{R}^{n \times d_2}$.

4.2 Time-dependent Multi-modal Interaction

To better capture the cross-modal semantic correspondences (Wu et al., 2020; Zhang et al., 2020), we design a long- and short-term hybrid memory gating (LSTHMG) block, which is a extension of standard LSTM.

We first obtain the current memory of each character-level representation for both modalities.

$$\hat{h}_i^x, c_i^x = \text{LSTM}_i^x(x_i, h_{i-1}^x, c_{i-1}^x) \quad (1)$$

$$\hat{h}_i^a, c_i^a = \text{LSTM}_i^a(a_i, h_{i-1}^a, c_{i-1}^a) \quad (2)$$

where LSTM denotes the standard LSTM (Graves et al., 2013).

After current updating, we employ multi-attention to control the different contributions of each hidden state.

$$h_i = \hat{h}_i + \text{MA}(\hat{h}_i^x, \hat{h}_i^a) \quad (3)$$

$$= \hat{h}_i + \sum_{l=0}^{L} (\text{softmax}(\frac{Q^l (K^l)^\top}{\sqrt{d}}) V^l) \quad (4)$$

where MA denotes the multi-attention gating mechanism, which is considered to mine multiple potential dimension-aware importance for each modality (Zadeh et al., 2018). $\hat{h}_i \in \mathbb{R}^{(d_1+d_2)\times 1}$ is the unsqueezed concatenation of $\hat{h}_i^x$ and $\hat{h}_i^a$. L denotes the max times for attentions. The query Q^l, key K^l and value V^l at the l-th time are defined similarly to self-attention (Vaswani et al., 2017):

$$Q^l = \hat{h}_i W_q^l, K^l = \hat{h}_i W_k^l, V^l = \hat{h}_i W_v^l \quad (5)$$

Note that h_i denotes the sum of L times attentional state concatenation for multi-modal representation at character-level step i, which is then used to perform word segmentation by CRF. Besides, we split each part for each modality as its own dimension: h_i^x and h_i^a, and input them into the next LSTHMG step.

Model	50		100		150	
	F	R_{oov}	F	R_{oov}	F	R_{oov}
BC(Text)	93.13	93.15	94.29	95.19	95.29	96.15
BC(Audio)	30.26	36.56	34.63	37.11	33.65	36.75
BC(Text+Audio)	34.43	37.54	32.67	36.68	33.39	37.04
WMSEG(Text)	94.26	94.25	95.24	95.47	95.39	95.13
WMSEG(Audio)	69.34	70.29	70.46	71.00	71.20	77.17
WMSEG(Text+Audio)	63.29	53.21	69.71	69.20	70.37	70.44
TMIN(Ours)	**94.72**	**94.28**	**95.96**	**95.84**	**96.62**	**96.73**

Table 2: Performance (the overall F-score and the recall of OOV) comparison of different approaches on different training size. We perform a Friedman test on model- (row-) wise p-value< 0.05.

4.3 CRF Labeling

Since the textual and acoustic semantics of each character have been integrated by time-dependent uni-modal and multi-modal interactions, we allow h_i to perform conditional sequence labeling. Instead of decoding each label independently, we model them jointly using a CRF to consider the correlations between labels in neighborhoods. Formally,

$$p(y|\hat{X}) = \frac{\prod_{i=1}^{n} \mathcal{S}_i(y_{i-1}, y_i, \hat{X})}{\sum_{y' \in Y} \prod_{i=1}^{n} \mathcal{S}_i(y'_{i-1}, y'_i, \hat{X})} \quad (6)$$

where $\mathcal{S}_i(y_{i-1}, y_i, \hat{X})$ and $\mathcal{S}_i(y'_{i-1}, y'_i, \hat{X})$ are potential functions. $\hat{X}$ denotes the input of CRF. Y denotes the output label space.

We use the maximum conditional likelihood estimation for CRF training. The logarithm of likelihood is given by: $\sum_i \log p(y|\hat{X})$. In the inference phase, we predict the output sequence that obtains the maximum score given by: $\text{argmax}_{y' \in Y} p(y|\hat{X})$.

5 Experimentation

In this section, we provide the exploratory experimental results and a case analysis.

5.1 Experimental Setting

Data Split. We evaluate our approach on the different size of training sets and the same validation set and test set, i.e., 50, 100 and 150 sentences for training, the remaining 50 and 50 sentences for validation and test, respectively. For different training sets, the Out-of-vocabulary (OOV) rate in test set is 92.89%, 46.73% and 30.93%, respectively.

Implementation Details. The character embeddings of text X are initialized with the cased BERT_{base} model pre-trained with dimension of 768, and fine-tuned during training. The character-level embeddings of audio A are encoded by Transformer with dimension of 124. The learning rate, the dropout rate, and the tradeoff parameter are respectively set to 1e-4, 0.5, and 0.5, which can achieve the best performance on the development set of both datasets via a small grid search over the combinations of [1e-5, 1e-4], [0.1, 0.5], and [0.1, 0.9] on two pieces of NVIDIA GTX 2080Ti GPU with pytorch 1.7. Based on best-performed development results, the Transformer layers for audio encoding and the multi-attention times L in gating is set 2 and 4, respectively. To motivate future research, the dataset, aligned features and code will be released [3].

Baselines. For a thorough comparison, we implement the following approaches with F1 as metric: 1) BERT and CRF framework, **BC**: **BC(Text)** , **BC(Audio)**, and **BC(Text+Audio)**. 2) A representative state-of-the-art, **WMSEG** (Tian et al., 2020b): **WMSEG(Text)**, **WMSEG(Audio)**, and **WMSEG(Text+Audio)**. Note that the approaches with (Text) take character-level text as input, the approaches with (Audio) take character-level audio as input, and the approaches with (Text+Audio) take character-level concatenation of text and audio as input.

5.2 Main Results

Table 2 shows the performance of different baselines compared with our approach, where the overall F-score and the recall of OOV are reported. From this table we can see that:

1) **WMSEG** performs much better than the general framework **BC**. This indicates that it is effective for **WMSEG** to incorporate wordhood information with several popular encoder-decoder com-

[3] https://github.com/MANLP-suda/MCWS

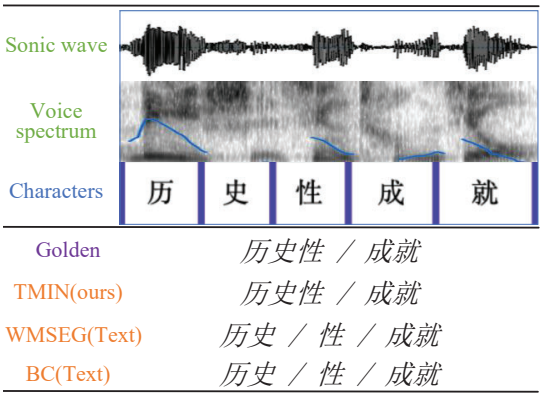


Figure 3: A case of the predicted results by different approaches. 1) 历史性(historic) 2) 历史(history), 性(sex) 3)成就(achievement).

binations and it is suitable as a competitive baseline.

2) The approach with only audio perform significantly worse than the approaches with text only, suggesting that it is confusing of the various acoustic features and we should utilize audio modality properly.

3) In most cases, the baselines with both text and audio bring in a poor performance compared with uni-modal approach, which suggests that simply concatenation of time-dependent character-level features for CWS seems a bad choice.

4) Among all approaches, our **TMIN** performs best, and significantly outperforms the competitive baselines (p-value<0.05). Moreover, with regard to R_{oov}, we can observe that our **TMIN** is able to recognize new words more accurately. This is mainly because our approach can obtain effective multi-modal information by time-dependent fusion against only textual, acoustic or early fused approaches.

5.3 Case Study

Figure 3 illustrates a real instance of the predicted boundaries by different approaches. From this figure, we can see that both **WMSRG** and **BC** give the wrong prediction of the boundary in "史" and "性" though they determine the correct segmentation for "历史" and "成就". However, our **TMIN** achieves all exact segmentation of this instance. This is mainly because it is very effective for audio, where there are a continuous breathing in the character " 性", thus "历史性" is a complete word.

6 Conclusion and Future Work

This paper proposes a new dataset for multi-modal Chinese word segmentation (MCWS), which is the first attempt to explore the multi-modality for traditional CWS. Meanwhile, we propose a time-dependent multi-modal interactive network (TMIN) to effectively integrate textual and acoustic features. The preliminary experimental results and case analysis demonstrate the reliability of our motivation and the effectiveness of the proposed approach.

In the future, we will annotate more samples at the current setting, and collect new samples with more modalities, such as visual information in social media, monologues and dialogues with continuous front face. Moreover, we will employ the neural active learning approaches for MCWS to reduce the annotation and achieve the best performance.

Acknowledgments

We thank all anonymous reviewers for their helpful comments. This work was supported by the National Key R&D Program of China under Grant No. 2020AAA0108600 and the NSFC grant No. 62076176. This work was also supported by a project funded by China Postdoctoral Science Foundation No. 2020M681713.

References

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1197–1206.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Xiaobin Wang, and Haitao Zheng. 2020. Coupling distant annotation and adversarial training for cross-domain chinese word segmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6662–6671.

Sufeng Duan and Hai Zhao. 2020. Attention is all you need for chinese word segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3862–3872.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethinkcws: Is chinese word segmentation a solved task? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5676–5686.

Chen Gong, Zhenghua Li, Bowei Zou, and Min Zhang. 2020. Multi-grained chinese word segmentation with weakly labeled data. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2026–2036.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649.

Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 2019. Incorporating word attention into character-based word segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2699–2709.

A. K. V. Sai Jayram, V. Ramasubramanian, and T. V. Sreenivas. 2002. Robust parameters for automatic segmentation of speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002, May 13-17 2002, Orlando, Florida, USA*, pages 513–514.

Xincheng Ju, Dong Zhang, Junhui Li, and Guodong Zhou. 2020. Transformer-based label set generation for multi-modal multi-label emotion detection. In *Proceedings of the 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pages 512–520.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of chinese representations? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3242–3252.

Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for chinese word segmentation. *Comput. Linguistics*, 35(4):505–512.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art chinese word segmentation with bi-lstms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4902–4908.

Mairgup Mansur, Wenzhe Pei, and Baobao Chang. 2013. Feature-based neural language model and chinese word segmentation. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 1271–1277.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 498–502.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 293–303.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2020. A concise model for multi-criteria chinese word segmentation with transformer encoder. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2887–2897.

Alaa Ehab Sakran, Sherif Mahdy Abdou, Salah Eldeen Hamid, and Mohsen Rashwan. 2017. A review: Automatic speech segmentation. *International Journal of Computer Science and Mobile Computing*, 6(4):308–315.

Weiwei Sun and Jia Xu. 2011. Enhancing chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in*

Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 970–979.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8286–8296.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020b. Improving Chinese word segmentation with wordhood memory networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online. Association for Computational Linguistics.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6558–6569.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Chunqi Wang and Bo Xu. 2017. Convolutional neural network with word embeddings for chinese word segmentation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 163–172.

Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019a. Unsupervised learning helps supervised neural word segmentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7200–7207.

Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2019b. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7216–7223.

Liangqing Wu, Dong Zhang, Qiyuan Liu, Shoushan Li, and Guodong Zhou. 2020. Speaker personality recognition with multimodal explicit many2many interactions. In *Proceedings of IEEE International Conference on Multimedia and Expo, ICME 2020, London, UK, July 6-10, 2020*, pages 1–6.

Fei Xia. 2000. The segmentation guidelines for the penn chinese treebank (3.0).

Nianwen Xu. 2003. Chinese word segmentation as character tagging. *Int. J. Comput. Linguistics Chin. Lang. Process.*, 8(1).

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 839–849.

Amir Zadeh, Paul Pu Liang, Soujanya Poria, Prateek Vij, Erik Cambria, and Louis-Philippe Morency. 2018. Multi-attention recurrent network for human communication comprehension. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5642–5649.

Dong Zhang, Xincheng Ju, Junhui Li, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 2020. Multi-modal multi-label emotion detection with modality and label dependence. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3584–3593.

Dong Zhang, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 2019. Effective sentiment-relevant word selection for multi-modal sentiment analysis in spoken language. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 148–156.

Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013. Exploring representations from unlabeled data with co-training for chinese word segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 311–321.

Meishan Zhang, Guohong Fu, and Nan Yu. 2017. Segmenting chinese microtext: Joint informal-word detection and segmentation with neural networks. In

Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pages 4228–4234.

Xiaoyan Zhao, Min Yang, Qiang Qu, and Yang Sun. 2020. Improving neural chinese word segmentation with lexicon-enhanced adaptive attention. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1953–1956.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 647–657.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xin-Yu Dai, and Jiajun Chen. 2017. Word-context character embeddings for chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 760–766.

Jianing Zhou, Jingkang Wang, and Gongshen Liu. 2019. Multiple character embeddings for chinese word segmentation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*, pages 210–216.

A Mixture-of-Experts Model for Antonym-Synonym Discrimination

Zhipeng Xie
School of Computer Science
Fudan University, Shanghai, China
xiezp@fudan.edu.cn

Nan Zeng
School of Computer Science
Fudan University, Shanghai, China
19212010017@fudan.edu.cn

Abstract

Discrimination between antonyms and synonyms is an important and challenging NLP task. Antonyms and synonyms often share the same or similar contexts and thus are hard to make a distinction. This paper proposes two underlying hypotheses and employs the mixture-of-experts framework as a solution. It works on the basis of a divide-and-conquer strategy, where a number of localized experts focus on their own domains (or subspaces) to learn their specialties, and a gating mechanism determines the space partitioning and the expert mixture. Experimental results have shown that our method achieves the state-of-the-art performance on the task.

1 Introduction

Antonymy-synonymy discrimination (ASD) is a crucial problem in lexical semantics and plays a vital role in many NLP applications such as sentiment analysis, textual entailment and machine translation. Synonymy refers to semantically-similar words (having similar meanings), while antonymy indicates the oppositeness or contrastiveness of words (having opposite meanings). Although telling apart antonyms and synonyms looks simple on the surface, it actually poses a hard problem because of their interchangeable substitution.

A few research efforts have been devoted to computational solutions of ASD task, which comprises two mainstreams: *pattern-based* and *distributional* approaches. The underlying idea of pattern-based methods exists in that antonymous word pairs co-occur with each other in some antonymy-indicating lexico-syntactic patterns within a sentence (Roth and im Walde, 2014; Nguyen et al., 2017). In spite of their high precision, pattern-based methods suffer from limited recall owing to the sparsity of lexico-syntactic patterns and the lexical variations.

Distributional methods work on the basis of *distributional hypothesis* stating that "the words similar in meaning tend to occur in similar contexts" (Harris, 1954). Traditional distributional methods are based on discrete context vectors. Scheible et al. (2013) verified that using only the contexts of certain classes can help discriminate antonyms and synonyms. Santus et al. (2014) thought that synonyms are expected to have broader and more salient intersection of their top-K salient contexts than antonyms, and proposed an Average-Precision-based unsupervised measure.

With the advent of word embeddings as the continuous representations (Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Pennington et al., 2014), several neural methods have been proposed to elicit ASD-specific information from pretrained word embeddings in a supervised manner. Etcheverry and Wonsever (2019) used a siamese network to ensure the symmetric, reflexive and transitive properties of synonymy and a parasiamese network to model the antitransitivity of antonymy. Ali et al. (2019) projected word embeddings into the synonym and antonym subspaces respectively, and then trained a classifier on the features from these distilled subspaces, where the trans-transitivity of antonymy was taken into consideration.

This paper follows the distributional approach and studies the ASD problem on the basis of pretrained word embeddings. Two hypotheses underlie our method: (a) antonymous words tend to be similar on most semantic dimensions but be different on only a few salient dimensions; (b) the salient dimensions may vary significantly for different antonymies throughout the whole distributional semantic space. With respect to the hypothesis (b), we find that a tailored model of mixture-of-experts (MoE) (Jacobs et al., 1991) fits it well. The semantic space is divided into a number of subspaces, and each subspace has one specialized expert to elicit

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 558–564
August 1–6, 2021. ©2021 Association for Computational Linguistics

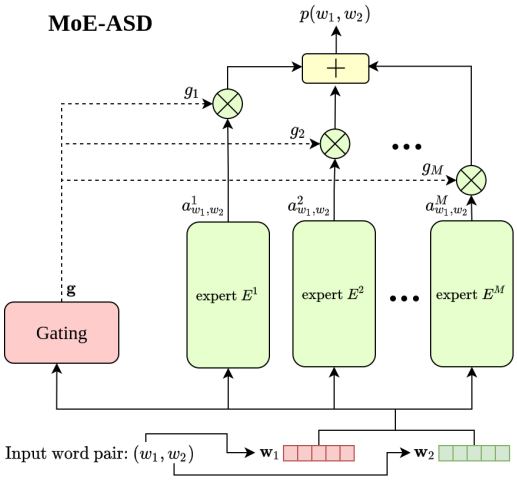


Figure 1: The architecture of MoE-ASD

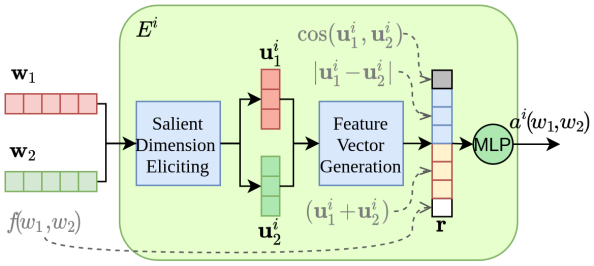


Figure 2: A localized expert

the salient dimensions and learn a discriminator for this subspace. As to the hypothesis (a), a similar opinion was also expressed by Cruse (1986) that antonymous words tend to have many common properties, but differ saliently along one dimension of meaning. In addition, our experimental results have shown that each expert requires only four salient dimensions to achieve the best performance.

Finally, we would like to point out the main difference of our method from the existing ones. Firstly, our MoE-ASD model adopts a divide-and-conquer strategy, where each subspace is in the charge of one relatively-simple localized expert that focuses on only a few salient dimensions; while existing methods rely on a global model which must grasp all the salient dimensions across all the subspaces. Secondly, our method simply enforces the symmetric property of synonymy and antonymy, but ignores the other algebraic properties such as the transitivity of synonymy and trans-transitivity of antonymy, because these algebraic properties do not always hold on the word level for the polysemy characteristic of words.

2 Method

This paper proposes a novel ASD method based on the mixture-of-experts framework (called **MoE-ASD**)[1]. Its architecture is illustrated in Figure 1. It solves the problem in a divide-and-conquer manner by dividing the problem space into a number of subspaces and each subspace is in the charge of a specialized expert. The expert focuses on the salient dimensions of the subspace and makes the decision for word pairs. A gating module is trained jointly with these experts. The details are as follows.

2.1 Localized Experts

All the experts are homogeneous, and they have the same network architecture but with different parameter values. Given a word pair (w_1, w_2) as input, each expert E^i computes its unnormalized probability $a^i(w_1, w_2)$ of being antonymy. As stated in Section 1, our method adopts the hypothesis that antonymous words tend to be similar on most semantic dimensions but be different on a few salient dimensions. Each expert has to first elicit the salient dimensions, and then makes a decision based on a feature vector constructed from them. Figure 2 illustrates how an expert works.

Let $\mathbf{w}_1$ and $\mathbf{w}_2$ denote the pre-trained word embeddings of words w_1 and w_2 respectively, whose dimensionality is d_e. Each expert E^i distills d_u salient dimensions from them by projecting them from $\mathbb{R}^{d_e}$ into $\mathbb{R}^{d_u}$:

$$\mathbf{u}^i_1 = \mathbf{w}_1 \cdot \mathbf{M}^i_u + \mathbf{b}^i_u \text{ and } \mathbf{u}^i_2 = \mathbf{w}_2 \cdot \mathbf{M}^i_u + \mathbf{b}^i_u \quad (1)$$

where $\mathbf{M}^i$ is a matrix of size $d_e \times d_u$ and $\mathbf{b}^i$ is a vector of length d_u. Next, a relational feature vector $\mathbf{r}$ is constructed by concatenating the sum $(\mathbf{u}^i_1 + \mathbf{u}^i_2)$, the absolute difference $|\mathbf{u}^i_1 - \mathbf{u}^i_2|$, the cosine similarity $\cos(\mathbf{u}^i_1, \mathbf{u}^i_2)$ and the prefix feature f_{w_1,w_2}:

$$\mathbf{r} = (\mathbf{u}^i_1 + \mathbf{u}^i_2) \oplus |\mathbf{u}^i_1 - \mathbf{u}^i_2| \oplus \cos(\mathbf{u}^i_1, \mathbf{u}^i_2) \oplus f_{w_1,w_2} \quad (2)$$

Here, f_{w_1,w_2} is the Negation-Prefix feature that denotes whether w_1 and w_2 differ only by one of the known negation prefixes: {*de*, *a*, *un*, *non*, *in*, *ir*, *anti*, *il*, *dis*, *counter*, *im*, *an*, *sub*, *ab*}, following Ali et al. (2019) and Rajana et al. (2017).

It is evident that the feature vector is symmetric with respect to the input word pair. This is,

[1] Our code and data are released at https://github.com/Zengnan1997/MoE-ASD

the word pairs (w_1, w_2) and (w_2, w_1) lead to the same feature vector. It is worth noting that the absolute difference is used instead of the difference, in order to preserve the symmetric properties of both synonymy and antonymy. We note that Roller et al. (2014) used the difference between two word vectors as useful features for detecting hypernymy which is asymmetric.

The relational feature vector $\mathbf{r}$ goes through an MLP to get the antonymy-score $a^i(w_1, w_2)$:

$$a^i(w_1, w_2) = (\mathbf{m}_o^i)^\top \cdot \text{ReLU}(\mathbf{r} \cdot \mathbf{M}_h^i + \mathbf{b}_h^i) + b_o^i \quad (3)$$

where the hidden layer has d_h units, $\mathbf{M}_h^i$ is a matrix of size $(2d_u + 2) \times d_h$, $\mathbf{b}_h^i$ and $\mathbf{m}_o^i$ are two vectors of length d_h, and b_o^i is the bias.

2.2 Gating Mechanism for Expert Mixture

Assume there are M localized experts in the MoE-ASD model. For an input word pair (w_1, w_2), we shall get M antonymy-scores $\mathbf{a} = [a^i(w_1, w_2)]_{1 \le i \le M}$, where each $a^i(w_1, w_2)$ is obtained from the expert E^i. Now, the problem is how to derive the final score for antonymy detection.

In our MoE-ASD model, the *final score* is a weighted average of the M scores from the localized experts:

$$s(w_1, w_2) = \mathbf{g}^\top \cdot \mathbf{a} \quad (4)$$

where $\mathbf{g}$ is located in the M-dimensional simplex, and denotes the proportional contributions of the experts to the final score. A gating mechanism is used to calculate $\mathbf{g}$ for each specific word pair (w_1, w_2), fulfilling a dynamic mixture of experts:

$$\mathbf{g} = \text{softmax}\left((\mathbf{w}_1 + \mathbf{w}_2)^\top \cdot \mathbf{M}_g\right) \quad (5)$$

where $\mathbf{M}_g \in \mathbb{R}^{d_e \times M}$ is the parameter matrix of the gating module. The i-th column of $\mathbf{M}_g$ can be thought of as the representative vector of the i-the expert, and the dot product between the sum of two word embeddings and the representative vector is the attention weight of the expert E^i. Softmax is then applied on the attention weights to get $\mathbf{g}$. It is evident that the gating module is also symmetric with respect to the input word pair. The symmetric properties of both the gating module and the local expert module endow our model with symmetry that make it distinct from the other state-of-the-arts such as **Parasiam** (Etcheverry and Wonsever, 2019) and **Distiller** (Ali et al., 2019).

Category	Train	Dev	Test	Total
Adjective	5562	398	1986	7946
Verb	2534	182	908	3624
Noun	2836	206	1020	4062

Table 1: Antonym/Synonym Dataset

2.3 Model Prediction and Loss Function

Given word pair (w_1, w_2), the probability of being antonymy is obtained by simply applying sigmoid function to the final score:

$$p(w_1, w_2) = \sigma(s(w_1, w_2)) \quad (6)$$

Let A denote the training set of N word pairs, $A = \{(w_1^{(n)}, w_2^{(n)})\}_{n=1}^N$, $t^{(n)}$ denote the gold label of the n-th word pair, and $p^{(n)}$ the predicted probability of being antonymy. Our model uses the cross-entropy loss function:

$$L = \frac{1}{N} \sum_{n=1}^{N} \left[t^{(n)} \log p^{(n)} + \left(1 - t^{(n)}\right) \log \left(1 - p^{(n)}\right) \right] \quad (7)$$

3 Evaluation

Dataset. We evaluate our method on the dataset (Nguyen et al., 2017) that was previously created from WordNet (Miller, 1995) and Wordnik[2]. The word pairs of antonyms and synonyms were grouped according to the word class (*Adjective*, *Noun* and *Verb*). The ratio of antonyms to synonyms in each group is 1:1. The statistics of the dataset are shown in Table 1. In order to make a fair comparison with previous algorithms, the dataset is splitted into training, validation and testing data the same as previous works.

Methods for Comparison: We make a comparison against the following ASD methods: (1) **Concat** - a baseline method that concatenates two word vectors and feeds it into an MLP with two hidden layers (with 400 and 200 hidden units respectively) and ReLU activation functions. (2) **AntSynNET** (Nguyen et al., 2017) is a pattern-based method that encodes the paths connecting the joint occurrences of candidate pairs using a LSTM; (3) **Parasiam** (Etcheverry and Wonsever, 2019) used a siamese network and a parasiamese network to ensure the algebraic properties of synonym and antonym, respectively. (4) **Distiller** (Ali et al., 2019) is a two-phase method that first distills

[2] http://www.wordnik.com

Method	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
Concat (Baseline)	0.596	0.751	0.651	0.596	0.750	0.656	0.688	0.745	0.708
AntSynNet (Nguyen et al., 2017)	0.750	0.798	0.773	0.717	0.826	0.768	0.807	0.827	0.817
Parasiam (Etcheverry and Wonsever, 2019)	0.855	0.857	0.856	0.864	**0.921**	0.891	0.837	0.859	0.848
Distiller (Ali et al., 2019)	0.854	**0.917**	0.884	0.871	0.912	0.891	0.823	0.866	0.844
MoE-ASD (Our method)	**0.878**	0.907	**0.892**	**0.895**	0.920	**0.908**	**0.841**	**0.900**	**0.869**

Table 2: Performance evaluation of our model and the baseline models (with vanilla word embeddings)

Method	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
AntSynNet (Nguyen et al., 2017)	0.763	0.807	0.784	0.743	0.815	0.777	0.816	0.898	0.855
Parasiam (Etcheverry and Wonsever, 2019)	0.874	**0.950**	0.910	0.837	**0.953**	0.891	0.847	0.939	0.891
Distiller (Ali et al., 2019)	0.912	0.944	0.928	0.899	0.944	0.921	0.905	0.918	0.911
MoE-ASD	**0.935**	0.941	**0.938**	**0.914**	0.944	**0.929**	**0.920**	**0.950**	**0.935**

Table 3: Performance evaluation with the dLCE embeddings

task-specific information and then trains a classifier based on distilled sub-spaces.

3.1 Experimental Settings

We use the 300-dimension FastText word embeddings (Bojanowski et al., 2017)[3]. The model is optimized with the Adam algorithm (Kingma and Ba, 2015). We run our algorithm 10 times and record the average Precision, Recall and F-scores. The number of salient dimensions (d_u) and the number of localized experts (M) are tuned on the validation data by grid search, with $M \in \{2^i\}_{1 \leq i \leq 8}$ and $d_u \in \{2^i\}_{1 \leq i \leq 8}$. The best configuration is ($d_u = 4, M = 256$) for both *Noun* and *Verb*, while ($d_u = 4, M = 128$) for *Adjective*.

3.2 Comparison with SOTA methods

Table 2 compares our method with the state-of-the-arts, which are restricted to pretrained vanilla word embeddings. Both the Parasiam method and our MoE-ASD method use FastText embeddings (Bojanowski et al., 2017), while *Distiller* uses Glove embeddings (Pennington et al., 2014).

It is observed that our model consistently outperforms the state-of-the-arts on all the three subtasks, which manifests the effectiveness of the mixture-of-experts model for ASD and validates the hypothesis (b) that the salient dimensions may vary significantly throughout the whole space.

We also find that the performance on *Noun* class is relatively low when compared with *Verb* and *Adjective* classes, which coincide with the observations obtained in (Scheible et al., 2013; Ali et al., 2019), possibly for the reason that polysemy phenomenon is more significant among nouns.

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip

F1-score	Adjective	Verb	Noun
Our full method	0.892	0.908	0.869
–prefix feature	0.886	0.905	0.868
–cosine sim	0.883	0.905	0.856
–absolute diff	0.890	0.897	0.866
–sum	0.888	0.903	0.867

Table 4: Ablation analysis of the features

Besides vanilla word embeddings, existing ASD methods also used dLCE (Nguyen et al., 2016) embeddings, and often obtained better results. However, a large number of antonymies and synonymies have been used in the process of learning dLCE embeddings, which may lead to severe overfitting. In spite of this concern, we also test our method with dLCE embeddings on the dataset and find that it outperforms these competitors with dLCE and list the results in Table 3.

3.3 Ablation Analysis of Features

We also make an ablation analysis about the four kinds of features, by removing each of them from our model. It can be seen from Table 4 that all the features are making their own contributions to the ASD. Different parts of speech have different sensitivities to different features. Specifically, verb is most sensitive to "*absolute difference*", while both adjective and noun are most sensitive to "*cosine*". The reason behind the observations deserves further exploration.

newdataset	Model	Adjective			Verb			Noun		
		P	R	F1	P	R	F1	P	R	F1
FastText	Parasiam	0.694	**0.866**	0.769	0.642	**0.824**	0.719	0.740	**0.759**	0.748
	MoE-ASD	**0.808**	0.810	**0.809**	**0.830**	0.693	**0.753**	**0.846**	0.722	**0.776**
dLCE	Parasiam	0.768	**0.952**	0.850	0.769	**0.877**	0.819	0.843	**0.914**	0.876
	MoE-ASD	**0.877**	0.908	**0.892**	**0.860**	0.835	**0.847**	**0.912**	0.869	**0.890**

Table 5: Performance of our model and the baseline models on the lexical-split datasets

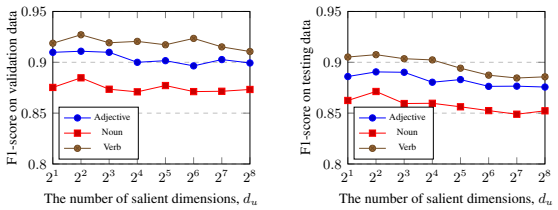


Figure 3: The effect on performance by varying the number of salient dimensions (fixing $M = 256$)

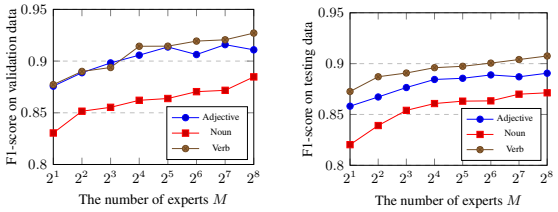


Figure 4: The effect on performance by varying the number of experts (fixing $d_u = 4$)

3.4 Hyperparameter Analysis

The number of salient dimensions (d_u) and the number of experts (M) are two prominent hyper-parameters in our MOE-ASD model. By varying their values, we study their influence on the performance.

Firstly, by fixing $M = 256$, we vary d_u from 2^1 to 2^8 and plot the F1-scores on the validation data and the testing data in Figure 3. It is observed that all the three subtasks (*Adjective*, *Noun* and *Verb*) arrive at the best performance at $d_u = 4$ on both validation data and testing data. It validates our hypothesis (a) that antonymous words tend to be different on only a few salient dimensions.

Secondly, by fixing $d_u = 4$, we vary M from 2^1 to 2^8 and plot the F1-scores in Figure 4. Overall, the performance becomes better with the larger number of experts. We conjecture that marginal improvement will be obtained by increasing the number of experts further, but we do not make such experiments.

Category	Train	Dev	Test	Total
Adjective	4227	303	1498	6028
Verb	2034	146	712	2892
Noun	2667	191	954	3812

Table 6: The datasets after lexical split

3.5 Lexical Memorization

To eliminate the bias introduced by the lexical memorization problem (Levy et al., 2015), we perform lexical splits to obtain train and test datasets with zero lexical overlap. The statistics of the lexical-split datasets are listed in Table 6. Table 5 shows the results of our method and Parasiam on the lexical-split datasets by using FastText and dLCE pretrained word embeddings. It can be seen that our MoE-ASD model outperforms Parasiam on all three lexical-split datasets. However, significant decreases in the F1 scores are also observed.

4 Conclusions

This paper first presents two hypotheses for ASD task (i.e., antonymous words tend to be different on only a few salient dimensions that may vary significantly for different antonymies) and then motivates an ASD method based on mixture-of-experts. Finally, experimental results have manifested its effectiveness and validated the two underlying hypotheses. It is worth noting that our method is distinct from the other state-of-the-arts in two main aspects: (1) it works in a *divide-and-conquer* strategy by dividing the whole space into multiple subspaces and having one expert specialized for each subspace; (2) it is *inherently symmetric* with respect to the input word pair.

Acknowledgments

This work is supported by National Key Research and Development Program of China (No.2018YFB1005100) and National Natural Science Foundation of China (No.62076072). We are grateful to the anonymous reviewers for their valuable comments.

References

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, the 31st Innovative Applications of Artificial Intelligence Conference, the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 6204–6211.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Benjamin Börschinger and Mark Johnson. 2011. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia.

Dvaid A Cruse. 1986. *Lexical Semantics*. Cambridge University Press.

Mathías Etcheverry and Dina Wonsever. 2019. Unraveling antonym's word vectors through a siamese-like network. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 3297–3307.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado. Association for Computational Linguistics.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2265–2273.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Distinguishing antonyms and synonyms in a pattern-based neural network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 76–85.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Sneha Rajana, Chris Callison-Burch, Marianna Apidianaki, and Vered Shwartz. 2017. Learning antonyms with paraphrases and a morphology-aware neural network. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, *SEM @ACM 2017, Vancouver, Canada, August 3-4, 2017*, pages 12–21. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1025–1036. ACL.

Michael Roth and Sabine Schulte im Walde. 2014. Combining word patterns and discourse markers for paradigmatic relation classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 524–530. The Association for Computer Linguistics.

Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. 2014. Taking antonymy mask off in vector space. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, pages 135–144.

Silke Scheible, Sabine Schulte im Walde, and Sylvia Springorum. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 489–497.

Learning Domain-Specialised Representations for Cross-Lingual Biomedical Entity Linking

Fangyu Liu, Ivan Vulić, Anna Korhonen, Nigel Collier
Language Technology Lab, TAL, University of Cambridge
{fl399, iv250, alk23, nhc30}cam.ac.uk

Abstract

Injecting external domain-specific knowledge (e.g., UMLS) into pretrained language models (LMs) advances their capability to handle specialised in-domain tasks such as biomedical entity linking (BEL). However, such abundant expert knowledge is available only for a handful of languages (e.g., English). In this work, by proposing a novel cross-lingual biomedical entity linking task (XL-BEL) and establishing a new XL-BEL benchmark spanning 10 typologically diverse languages, we first investigate the ability of standard knowledge-agnostic as well as knowledge-enhanced monolingual and multilingual LMs beyond the standard monolingual English BEL task. The scores indicate large gaps to English performance. We then address the challenge of transferring domain-specific knowledge from resource-rich languages to resource-poor ones. To this end, we propose and evaluate a series of cross-lingual transfer methods for the XL-BEL task, and demonstrate that general-domain bitext helps propagate the available English knowledge to languages with little to no in-domain data. Remarkably, we show that our proposed domain-specific transfer methods yield consistent gains across all target languages, sometimes up to 20 Precision@1 points, without any in-domain knowledge in the target language, and without any in-domain parallel data.

1 Introduction

Recent work has demonstrated that it is possible to combine the strength of 1) Transformer-based encoders such as BERT (Devlin et al., 2019; Liu et al., 2019), pretrained on large general-domain data with 2) external linguistic and world knowledge (Zhang et al., 2019; Levine et al., 2020; Lauscher et al., 2020). Such expert human-curated knowledge is crucial for NLP applications in specialised domains such as biomedicine. There, Liu et al. (2021) recently proposed *self-alignment pretraining* (SAP), a technique to fine-tune BERT on phrase-level synonyms extracted from the Unified Medical Language System (UMLS; Bodenreider 2004).[1] Their SAPBERT model currently holds state-of-the-art (SotA) across all major English biomedical entity linking (BEL) datasets. However, this approach is not widely applicable to other languages: abundant external resources are available only for a few languages, hindering the development of domain-specific NLP models in all other languages.

Simultaneously, exciting breakthroughs in cross-lingual transfer for language understanding tasks have been achieved (Artetxe and Schwenk, 2019; Hu et al., 2020). However, it remains unclear whether such transfer techniques can be used to improve domain-specific NLP applications and mitigate the gap between knowledge-enhanced models in resource-rich versus resource-poor languages. In this paper, we thus investigate the current performance gaps in the BEL task beyond English, and propose several cross-lingual transfer techniques to improve domain-specialised representations and BEL in resource-lean languages.

In particular, we first present a novel cross-lingual BEL (XL-BEL) task and its corresponding evaluation benchmark in 10 typologically diverse languages, which aims to map biomedical names/mentions in any language to the controlled UMLS vocabulary. After empirically highlighting the deficiencies of multilingual encoders (e.g, MBERT and XLMR; Conneau et al. 2020) on XL-BEL, we propose and evaluate a multilingual extension of the SAP technique. Our main results suggest that expert knowledge can be transferred from English to resource-leaner languages, yielding huge gains over vanilla MBERT and XLMR, and English-only SAPBERT. We also show that

[1] UMLS is a large-scale biomedical knowledge graph containing more than 14M biomedical entity names.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 565–574
August 1–6, 2021. ©2021 Association for Computational Linguistics

leveraging general-domain word and phrase translations offers substantial gains in the XL-BEL task.

Contributions. **1)** We highlight the challenge of learning (biomedical) domain-specialised cross-lingual representations. **2)** We propose a novel multilingual XL-BEL task with a comprehensive evaluation benchmark in 10 languages. **3)** We offer systematic evaluations of existing knowledge-agnostic and knowledge-enhanced monolingual and multilingual LMs in the XL-BEL task. **4)** We present a new SotA multilingual encoder in the biomedical domain, which yields large gains in XL-BEL especially on resource-poor languages, and provides strong benchmarking results to guide future work. The code, data, and pretrained models are available online at: `github.com/cambridgeltl/sapbert`.

2 Methodology

Background and Related Work. Learning biomedical entity representations is at the core of BioNLP, benefiting, e.g., relational knowledge discovery (Wang et al., 2018) and literature search (Lee et al., 2016). In the current era of contextualised representations based on Transformer architectures (Vaswani et al., 2017), biomedical text encoders are pretrained via Masked Language Modelling (MLM) on diverse biomedical texts such as PubMed articles (Lee et al., 2020; Gu et al., 2020), clinical notes (Peng et al., 2019; Alsentzer et al., 2019), and even online health forum posts (Basaldella et al., 2020). However, it has been empirically verified that naively applying MLM-pretrained models as entity encoders does not perform well in tasks such as biomedical entity linking (Basaldella et al., 2020; Sung et al., 2020). Recently, Liu et al. (2021) proposed SAP (**S**elf-**A**lignment **P**retraning), a fine-tuning method that leverages synonymy sets extracted from UMLS to improve BERT's ability to act as a biomedical entity encoder. Their SAPBERT model currently achieves SotA scores on all major English BEL benchmarks.

In what follows, we first outline the SAP procedure, and then discuss the extension of the method to include multilingual UMLS synonyms (§2.1), and then introduce another SAP extension which combines domain-specific synonyms with general-domain translation data (§2.2).

2.1 Language-Agnostic SAP

Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ denote the tuple of a name and its categorical label. When learning from UMLS synonyms, $\mathcal{X} \times \mathcal{Y}$ is the set of all *(name, CUI*[2]*)* pairs, e.g., (*vaccination*, `C0042196`). While Liu et al. (2021) use only English names, we here consider names in other UMLS languages. During training, the model is steered to create similar representations for synonyms regardless of their language.[3] The learning scheme includes 1) an online sampling procedure to select training examples and 2) a metric learning loss that encourages strings sharing the same CUI to obtain similar representations.

Training Examples. Given a mini-batch of N examples $\mathcal{B} = \mathcal{X}_\mathcal{B} \times \mathcal{Y}_\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{N}$, we start from constructing all possible triplets for all names $x_i \in \mathcal{X}_\mathcal{B}$. Each triplet is in the form of (x_a, x_p, x_n) where x_a is the *anchor*, an arbitrary name from $\mathcal{X}_\mathcal{B}$; x_p is a positive match of x_a (i.e., $y_a = y_p$) and x_n is a negative match of x_a (i.e., $y_a \neq y_n$). Let $f(\cdot)$ denote the encoder (i.e., MBERT or XLMR in this paper). Among the constructed triplets, we select all triplets that satisfy the following constraint:

$$\|f(x_a) - f(x_p)\|_2 + \lambda \geq \|f(x_a) - f(x_n)\|_2,$$

where λ is a predefined margin. In other words, we only consider triplets with the positive sample further to the negative sample by a margin of λ. These 'hard' triplets are more informative for representation learning (Liu et al., 2021). Every selected triplet then contributes one positive pair (x_a, x_p) and one negative pair (x_a, x_n). We collect all such positives and negatives, and denote them as $\mathcal{P}, \mathcal{N}$.

Multi-Similarity Loss. We compute the pairwise cosine similarity of all the name representations and obtain a similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{X}_\mathcal{B}| \times |\mathcal{X}_\mathcal{B}|}$ where each entry $\mathbf{S}_{ij}$ is the cosine similarity between the i-th and j-th names in the mini-batch $\mathcal{B}$. The Multi-Similarity loss (MS, Wang et al. 2019), is then used for learning from the triplets:

$$\mathcal{L} = \frac{1}{|\mathcal{X}_\mathcal{B}|} \sum_{i=1}^{|\mathcal{X}_\mathcal{B}|} \left(\frac{1}{\alpha} \log \left(1 + \sum_{n \in \mathcal{N}_i} e^{\alpha(\mathbf{S}_{in} - \epsilon)} \right) + \frac{1}{\beta} \log \left(1 + \sum_{p \in \mathcal{P}_i} e^{-\beta(\mathbf{S}_{ip} - \epsilon)} \right) \right). \quad (1)$$

α, β are temperature scales; ϵ is an offset applied on the similarity matrix; $\mathcal{P}_i, \mathcal{N}_i$ are indices of positive and negative samples of the i-th *anchor*.

[2] In UMLS, "CUI" means **C**oncept **U**nique **I**dentifier.

[3] For instance, *vaccination* (EN), *active immunization* (EN), *vacunación* (ES) and 予防接種 (JA) all share the same Concept Unique Identifier (CUI; `C0042196`); thus, they should all have similar representations.

#↓, language→	EN	ES	DE	FI	RU	TR	KO	ZH	JA	TH
sentences	-	223,506	350,193	77,736	206,060	29,473	47,702	136,054	157,670	19,066
unique titles (Wiki page)	60,598	37,935	24,059	15,182	21,044	5,251	10,618	17,972	11,002	4,541
mentions	1,067,083	204,253	431,781	105,182	221,383	29,958	60,979	197,317	220,452	31,177
unique mentions	121,669	25,169	44,390	26,184	28,302	4,110	9,032	24,825	21,949	5,064
unique mentions$_{\text{mention!=title}}$	69,199	22,162	43,753	19,409	23,935	2,833	3,740	12,046	12,571	2,480

Table 1: Construction of the XL-BEL benchmark; key statistics. See the App. §A.1 for further details.

2.2 SAP with General-Domain Bitext

We also convert word and phrase translations into the same format (§2.1), where each 'class' now contains only two examples. For a translation pair (x_p, x_q), we create a unique pseudo-label y_{x_p,x_q} and produce two new name-label instances (x_p, y_{x_p,x_q}) and (x_q, y_{x_p,x_q}),[4] and proceed as in §2.1. This allows us to easily combine domain-specific knowledge with general translation knowledge within the same SAP framework.

3 The XL-BEL Task and Evaluation Data

A general cross-lingual entity linking (EL) task (McNamee et al., 2011; Tsai and Roth, 2016) aims to map a mention of an entity in free text of *any language* to a controlled English vocabulary, typically obtained from a knowledge graph (KG). In this work, we propose XL-BEL, a cross-lingual *biomedical* EL task. Instead of grounding entity mentions to English-specific ontologies, we use UMLS as a language-agnostic KG: the XL-BEL task requires a model to associate a mention in any language to a (language-agnostic) CUI in UMLS. XL-BEL thus serves as an ideal evaluation benchmark for biomedical entity representations: it challenges the capability of both 1) representing domain entities and also 2) associating entity names in different languages.

Evaluation Data Creation. For English, we take the available BEL dataset WikiMed (Vashishth et al., 2020), which links Wikipedia mentions to UMLS CUIs. We then follow similar procedures as WikiMed and create an XL-BEL benchmark covering 10 languages (see Table 2). For each language, we extract all sentences from its Wikipedia dump, find all hyperlinked concepts (i.e., words and phrases), lookup their Wikipedia pages, and retain only concepts that are linked to UMLS.[5] For each UMLS-linked mention, we add a triplet *(sentence, mention, CUI)* to our dataset.[6] Only one example per surface form is retained to ensure diversity. We then filter out examples with mentions that have the same surface form as their Wikipedia article page.[7] Finally, 1k examples are randomly selected for each language: they serve as the final test sets in our XL-BEL benchmark. The statistics of the benchmark are available in Table 1.

4 Experiments and Results

UMLS Data. We rely on the UMLS (2020AA) as our SAP fine-tuning data, leveraging synonyms in all available languages. The full multilingual fine-tuning data comprises ≈15M biomedical entity names associated with ≈4.2M individual CUIs. As expected, English is dominant (69.6% of all 15M names), followed by Spanish (10.7%) and French (2.2%). The full stats are in App. §A.3.

Translation Data. We use (a) "muse" word translations (Lample et al., 2018), and (b) the parallel Wikipedia article titles (phrase-level translations; referred to as "wt"). We also list results when using "muse" and "wt" combined ("wt+ muse").

Training and Evaluation Details. Our SAP fine-tuning largely follows Liu et al. (2021); we refer to the original work and the Appendix for further tech-

[4] These pseudo-labels are not related to UMLS, but are used to format our parallel translation data into the input convenient for the SAP procedure. In practice, for these data we generate pseudo-labels ourselves as 'LANGUAGE_CODE+index'. For instance, `ENDE2344` indicates that this word pair is our 2,344th English-German word translation. Note that the actual coding scheme does not matter as it is only used for our algorithm to determine what terms belong to the same (in this case - translation) category.

[5] For instance, given a sentence from German Wikipedia *Die [Inkubationszeit] von COVID-19 beträgt durchschnittlich fünf bis sechs Tage.*, we extract the hyperlinked word *Inkubationszeit* as an UMLS-linked entity mention. Since Wikipedia is inherently multilingual, if *Inkubationszeit* is linked to UMLS, its cross-lingual counterparts, e.g., *Incubation period* (EN), are all transitively linked to UMLS.

[6] Note that though each mention is accompanied with its context, we regard it as out-of-context mention following the tradition in prior work (Sung et al., 2020; Liu et al., 2021; Tutubalina et al., 2020). According to Basaldella et al. (2020), biomedical entity representations can be easily polluted by its context. We leave contextual modelling for future work.

[7] Otherwise, the problem is easily solved by comparing surface forms of the mention and the article title.

language→	EN		ES		DE		FI		RU		TR		KO		ZH		JA		TH		**avg**	
model↓	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
monolingual models																						
{$LANG}BERT	-	-	41.3	42.5	16.8	18.4	4.9	5.2	1.1	1.6	19.5	21.8	1.1	1.6	2.1	3.2	2.7	2.8	0.4	0.4	10.0	10.8
+ SAP$_{all_syn}$	-	-	60.9	66.8	**35.5**	**40.0**	**18.8**	**23.9**	**36.4**	**42.4**	**44.9**	**49.7**	13.5	16.0	18.5	**23.8**	21.2	25.9	0.6	0.6	27.8	32.1
SAPBERT	**78.7**	**81.6**	47.3	51.4	22.7	24.7	8.2	10.2	5.8	6.0	26.4	29.7	2.0	2.4	1.9	2.2	3.0	3.2	3.1	3.4	19.9	21.6
SAPBERT$_{all_syn}$	78.3	80.7	55.6	61.3	30.0	34.2	11.8	14.8	9.3	11.3	35.5	39.5	2.0	2.4	6.4	8.2	6.9	8.3	3.0	3.3	23.9	26.4
multilingual models																						
MBERT	0.8	1.7	0.5	0.7	0.3	0.4	0.4	0.8	0.0	0.0	0.7	1.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.3	0.5
+ SAP$_{en_syn}$	75.5	79.9	50.6	55.8	26.0	29.6	8.7	10.7	10.1	12.6	31.0	34.4	2.7	3.2	4.1	5.7	4.7	5.9	3.1	3.5	21.7	24.1
+ SAP$_{all_syn}$	75.0	79.7	**61.4**	**67.0**	33.4	37.8	18.4	21.9	35.1	40.3	44.5	47.7	15.1	17.6	**19.5**	22.7	19.9	25.0	2.8	3.4	32.5	36.3
XLMR	1.0	2.0	0.3	0.7	0.0	0.1	0.1	0.2	0.1	0.2	0.4	0.5	0.0	0.3	0.1	0.2	0.2	0.4	0.0	0.1	0.2	0.5
+ SAP$_{en_syn}$	78.1	80.9	47.9	53.5	27.6	32.0	12.2	14.7	21.8	25.9	29.3	35.9	4.5	6.7	7.9	11.3	8.3	11.3	11.5	16.2	24.9	28.8
+ SAP$_{all_syn}$	78.2	81.0	56.4	62.7	31.8	37.3	18.6	22.2	35.4	41.2	42.8	48.9	**16.7**	**21.4**	18.8	23.0	**24.0**	**28.1**	**20.6**	**27.5**	**34.3**	**39.3**

Table 2: Various base models combined with SAP, using either all synonyms (*all_syn*) or only English synonyms (*en_syn*) in UMLS. {$LANG} denotes the language of the corresponding column (also in Table 4). See Table 6 (App. §A.3) for the language codes. **avg** refers to the average performance across all target languages. Grey and light blue rows are off-the-shelf base models and models fine-tuned with the UMLS knowledge, respectively.

language→	ES		DE		FI		RU		TR		KO		ZH		JA		TH		**avg**	
model↓	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
XLMR + SAP$_{en_syn}$	47.9	53.5	27.6	32.0	12.2	14.7	21.8	25.9	29.3	35.9	4.5	6.7	7.9	11.3	8.3	11.3	11.5	16.2	19.0	23.1
+ en-{$LANG} wt	55.0	62.2	34.6	41.4	18.6	24.4	35.0	41.5	43.3	50.6	15.9	22.3	15.9	23.0	18.7	24.4	25.1	32.4	29.1	35.8
+ en-{$LANG} muse	54.4	61.0	28.7	34.4	16.7	20.6	33.6	39.0	41.9	48.8	11.9	16.3	12.3	16.7	15.7	19.9	18.6	25.1	26.0	31.3
+ en-{$LANG} wt+muse	49.4	59.6	30.3	36.9	20.4	28.9	33.2	41.9	42.7	51.7	16.1	22.3	16.0	22.9	17.8	24.3	26.2	34.0	28.0	35.8
XLMR + SAP$_{all_syn}$	56.4	62.7	31.8	37.3	18.6	22.2	35.4	41.2	42.8	48.9	16.7	21.4	18.8	23.0	24.0	28.1	20.6	27.5	29.5	34.7
+ en-{$LANG} wt	57.2	63.7	35.1	42.3	20.3	27.6	35.8	43.8	48.8	55.0	22.1	27.9	20.6	27.3	24.8	**31.3**	30.0	37.6	32.7	**39.6**
+ en-{$LANG} muse	57.9	63.9	33.0	38.4	23.0	27.3	39.8	45.9	47.2	54.5	22.1	25.7	19.2	25.6	**25.2**	30.2	25.9	32.8	32.6	38.3
+ en-{$LANG} wt+ muse	51.4	61.2	31.3	38.9	22.8	28.4	36.4	45.2	42.2	51.6	**24.4**	**29.2**	21.1	28.2	23.2	30.4	**30.9**	**37.9**	31.5	39.0
MBERT + SAP$_{all_syn}$	**61.4**	67.0	33.4	37.8	18.4	21.9	35.1	40.3	44.5	47.7	15.1	17.6	19.5	22.7	19.9	25.0	2.8	3.4	27.8	31.5
+ en-{$LANG} wt	59.2	66.9	**37.5**	**43.9**	25.6	33.0	39.6	47.2	52.7	59.7	19.8	24.3	24.1	31.9	23.5	28.7	4.8	5.9	31.9	37.9
+ en-{$LANG} muse	59.9	66.2	34.3	38.8	21.6	27.5	36.5	41.7	51.0	56.7	18.1	21.2	22.2	26.4	22.0	25.5	3.4	3.8	29.2	34.2
+ en-{$LANG} wt+ muse	59.2	**67.5**	35.3	42.4	**30.5**	**37.3**	**41.6**	**49.2**	**57.2**	**64.7**	19.8	25.0	**24.6**	**32.1**	24.3	28.0	5.2	6.3	**33.1**	39.2

Table 3: Results when applying SAP with 1) UMLS knowledge + 2) word and/or phrase translations.

nical details. The evaluation measure is standard Precision@1 and Precision@5. In all experiments, SAP always denotes fine-tuning of a base LM with UMLS data. `[CLS]` of the last layer's output is used as the final representation (Liu et al., 2021). Without explicit mentioning, we use the BASE variants of all monolingual and multilingual LMs. At inference, given a query representation, a nearest neighbour search is used to rank all candidates' representations. We restrict the target ontology to only include CUIs that appear in WikiMed (62,531 CUIs, 399,931 entity names).

4.1 Main Results and Discussion

Multilingual UMLS Knowledge Always Helps (Table 2). Table 2 summarises the results of applying multilingual SAP fine-tuning based on UMLS knowledge on a wide variety of monolingual, multilingual, and in-domain pretrained encoders. Injecting UMLS knowledge is consistently beneficial to the models' performance on XL-BEL across all languages and across all base encoders. Using multilingual UMLS synonyms to SAP-fine-tune the biomedical PUBMEDBERT (SAPBERT$_{all_syn}$) instead of English-only synonyms (SAPBERT) improves its performance across the board. SAP-ing monolingual BERTs for each language also yields substantial gains across all languages; the only exception is Thai (TH), which is not represented in UMLS. Fine-tuning multilingual models MBERT and XLMR leads to even larger relative gains.

Performance across Languages (Table 2). UMLS data is heavily biased towards Romance and Germanic languages. As a result, for languages more similar to these families, monolingual LMs (upper half, Table 2) are on par or outperform multilingual LMs (lower half, Table 2). However, for other (distant) languages (e.g., KO, ZH, JA, TH), the opposite holds. For instance, on TH, XLMR+SAP$_{all_syn}$ outperforms THBERT+SAP$_{all_syn}$ by 20% Precision@1.

General Translation Knowledge is Useful (Table 3). Table 3 summarises the results where we

language→	ES		DE		RU		KO		**avg**	
model↓	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
MBERT										
+ SAP$_{en_syn}$	50.6	55.8	26.0	29.6	10.1	12.6	2.7	3.2	22.4	25.3
+ SAP$_{\{\$LANG\}_syn}$	57.1	62.8	28.9	33.6	25.8	31.7	2.1	2.6	28.5	32.7
+ SAP$_{en+\{\$LANG\}_syn}$	61.1	**68.5**	**35.2**	**39.8**	**35.6**	**40.9**	14.4	16.3	**36.6**	**41.4**
+ SAP$_{all_syn}$	**61.4**	67.0	33.4	37.8	35.1	40.3	**15.1**	**17.6**	36.6	40.7
XLMR										
+ SAP$_{en_syn}$	47.9	53.5	27.6	32.0	21.8	25.9	4.5	6.7	25.5	29.5
+ SAP$_{\{\$LANG\}_syn}$	52.9	55.8	25.9	30.4	28.7	34.2	2.4	2.9	24.5	30.8
+ SAP$_{en+\{\$LANG\}_syn}$	55.8	62.5	27.7	32.3	**36.4**	**42.2**	15.8	19.8	33.9	39.2
+ SAP$_{all_syn}$	**56.4**	**62.7**	**31.8**	**37.3**	35.4	41.2	**16.7**	**21.4**	**35.1**	**40.7**

Table 4: Varying UMLS synonymy sets.

continue training on general translation data (§2.2) after the previous UMLS-based SAP. With this variant, base multilingual LMs become powerful multilingual biomedical experts. We observe additional strong gains (cf., Table 2) with out-of-domain translation data: e.g., for MBERT the gains range from 2.4% to 12.7% on all languages except ES. For XLMR, we report Precision@1 boosts of >10% on RU, TR, KO, TH with XLMR+SAP$_{en_syn}$, and similar but smaller gains also with XLMR+SAP$_{all_syn}$.

We stress the case of TH, not covered in UMLS. Precision@1 rises from 11.5% (XLMR+SAP$_{en_syn}$) to 30.9%$^{\uparrow 19.4\%}$ (XLMR+SAP$_{all_syn}$(+en-th wt+ muse)), achieved through the synergistic effect of both knowledge types: **1) UMLS synonyms in other languages** push the scores to 20.6%$^{\uparrow 9.1\%}$; **2) translation knowledge** increases it further to 30.9%$^{\uparrow 10.3\%}$. In general, these results suggest that both external in-domain knowledge and general-domain translations boost the performance in resource-poor languages.

The More the Better (Table 4)? According to Table 4 (lower half), it holds almost universally that all_syn > en+{$LANG}_syn > en_syn/{$LANG}_syn on XLMR, that is, it seems that more in-domain knowledge (even in non-related languages) benefit cross-lingual transfer. However, for MBERT (Table 4, upper half), the trend is less clear, with en+{$LANG}_syn sometimes outperforming the all_syn variant. Despite modest performance differences, this suggests that the choice of source languages for knowledge transfer also plays a role; this warrants further investigations in future work.

Are Large Models (Cross-Lingual) Domain Experts (Table 5)? We also investigate the LARGE variant of XLMR, and compare it to its BASE variant. On English, XLMR$_{LARGE}$ gets 73.0% Precision@1, being in the same range as SAPBERT (78.7%), without SAP-tuning (Table 5). The scores without SAP fine-tuning on XLMR$_{LARGE}$, although much higher than of its BASE variant, decrease on other ('non-English') languages. At the same time, note that XLMR BASE achieves random-level performance without SAP-tuning. After SAP fine-tuning, on average, XLMR$_{LARGE}$+SAP still outperforms BASE models, but the gap is much smaller: e.g., we note that the performance of the two SAP-ed models is on par in English. This suggests that with sufficient knowledge injection, the underlying base model is less important (English); however, when the external data are scarce (other languages beyond English), a heavily parameterised large pretrained encoder can boost knowledge transfer to resource-poor languages.

data split→	EN		**avg**	
model↓	@1	@5	@1	@5
XLMR	1.0	2.0	0.2	0.5
+ SAP$_{all_syn}$	78.2	81.0	34.3	39.3
XLMR$_{LARGE}$	73.0	75.0	12.3	13.3
+ SAP$_{all_syn}$	**78.3**	**81.3**	**39.0**	**44.2**

Table 5: Comparing BASE and LARGE models on XL-BEL. Both EN results and **avg** across all languages are reported. Full table available in Appendix Table 9.

5 Conclusion

We have introduced a novel cross-lingual biomedical entity task (XL-BEL), establishing a wide-coverage and reliable evaluation benchmark for cross-lingual entity representations in the biomedical domain in 10 languages, and have evaluated current SotA biomedical entity representations on XL-BEL. We have also presented an effective transfer learning scheme that leverages general-domain translations to improve the cross-lingual ability of domain-specialised representation models. We hope that our work will inspire more research on multilingual *and* domain-specialised representation learning in the future.

Acknowledgements

We thank the three reviewers and the AC for their insightful comments and suggestions. FL is supported by Grace & Thomas C.H. Chan Cambridge Scholarship. IV and AK are supported by the ERC Consolidator Grant LEXICAL (no. 648909) awarded to AK. NC kindly acknowledges grant-in-aid support from the UK ESRC for project EPI-AI (ES/T012277/1).

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Marco Basaldella, Fangyu Liu, Ehsan Shareghi, and Nigel Collier. 2020. COMETA: A corpus for medical entity linking in the social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3122–3137, Online. Association for Computational Linguistics.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *arXiv:2007.15779*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2020. Specializing unsupervised pretraining models for word-level semantic similarity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1371–1383, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Sunwon Lee, Donghyeon Kim, Kyubum Lee, Jaehoon Choi, Seongsoon Kim, Minji Jeon, Sangrak Lim, Donghee Choi, Sunkyu Kim, Aik-Choon Tan, et al. 2016. BEST: next-generation biomedical entity search tool for knowledge discovery from biomedical literature. *PloS one*, 11(10):e0164680.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas Oard, and David Doermann. 2011. Cross-

language entity linking. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 255–263, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. 2020. Biomedical entity representations with synonym marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3641–3650, Online. Association for Computational Linguistics.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598, San Diego, California. Association for Computational Linguistics.

Elena Tutubalina, Artur Kadurin, and Zulfat Miftahutdinov. 2020. Fair evaluation in concept normalization: a large-scale comparative analysis for BERT-based models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6710–6716, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shikhar Vashishth, Rishabh Joshi, Denis Newman-Griffis, Ritam Dutt, and Carolyn Rose. 2020. MedType: Improving Medical Entity Linking with Semantic Type Prediction. *arXiv e-prints*, page arXiv:2005.00460.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5022–5030. Computer Vision Foundation / IEEE.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. 2018. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, 87:12–20.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Appendix A

A.1 XL-BEL: Full Statistics

Table 1 in the main paper summarises the key statistics of the XL-BEL benchmark. It was extracted from the `20200601` version of Wikipedia dump. "sentences" refers to the number of sentences that contain biomedical mentions in the Wiki dump. "unique titles (Wiki page)" denotes the number of unique Wikipedia articles the biomedical mentions link to. "mentions" denotes the number of all biomedical mentions in the Wikipedia dump. "unique mentions" refers to the number of mentions after filtering out examples containing duplicated mention surface forms. "unique $\text{mentions}_{\text{mention!=title}}$" denotes the number of unique mentions that have surface forms different from the Wikipedia articles they link to. The 1k test sets for each language are then randomly selected from the examples in "unique $\text{mentions}_{\text{mention!=title}}$".

A.2 XL-BEL: Selection of Languages

Our goal is to select a diverse and representative sample of languages for the resource and evaluation from the full set of possibly supported languages. For this reason, we exclude some Romance and Germanic languages which were too similar to some languages already included in the resource (e.g., since we include Spanish as a representative of the Romance language, evaluating on related languages such as Portuguese or Italian would not yield additional and new insights, while it would just imply running additional experiments). The language list covers languages that are close to English (Spanish, German); languages that are very distant from English (Thai, Chinese, etc.); and also languages that are *in the middle* (e.g., Turkish, which is typologically different, but shares a similar writing script with English).

The availability of biomedical texts in Wikipedia also slightly impacted our choice of languages. The overlapping entities of Wikipedia and UMLS are not evenly distributed in the biomedical domain. For example, since animal species are comprehensively encoded in UMLS, they become rather dominant for certain low-resource languages. We manually inspected the distribution of the covered entities in each language to ensure that they are indeed representative biomedical concepts. Languages with heavily skewed entity distributions are filtered out. E.g., biomedical concepts in Basque Wikipedia are heavily skewed towards plant and animal species (which are valid UMLS concepts but not representative enough). As a result, we dropped Basque as our evaluation language. The current 10 languages all have a reasonably fair distribution over biomedical concepts categories.

A.3 UMLS Data Preparation

All our UMLS fine-tuning data for SAP is extracted from the `MRCONSO.RRF` file downloaded at https://www.nlm.nih.gov/research/umls/licensedcontent/umlsarchives04.html#2020AA. The extracted data includes 147,706,62 synonyms distributed in more than 20 languages. The detailed statistics are available in Table 6.

code	language	# synonyms	percentage
EN	English	10,277,246	69.6%
ES	Spanish	1,575,109	10.7%
JA	Japanese	329,333	2.2%
RU	Russian	291,554	2.0%
DE	German	231,098	1.6%
KO	Korean	145,865	1.0%
ZH	Chinese	80,602	0.5%
TR	Turkish	51,328	0.3%
FI	Finnish	24,767	0.2%
TH	Thai	0	0.0%
FR	French	428,406	2.9%
PT	Portuguese	309,448	2.1%
NL	Dutch	290,415	2.0%
IT	Italian	242,133	1.3%
CS	Czech	196,760	0.7%
NO	Norwegian	63,075	0.4%
PL	Polish	51,778	0.4%
ET	Estonian	31,107	0.2%
SV	Swedish	29,716	0.2%
HR	Croatian	10,035	0.1%
EL	Greek	2,281	<0.1%
LV	Latvian	1,405	<0.1%
	Total	147,706,62	100%

Table 6: The amount of UMLS synonyms per language. The first 10 languages are included in our XL-BEL test languages. However, note that Thai has no UMLS data.

A.4 Translation Data

The full statistics of the used word and phrase translation data are listed in Table 7. The "muse" word translations are downloaded from https://github.com/facebookresearch/MUSE while the Wikititle pairs ("wt") are extracted by us, and are made publicly available.

A.5 Pretrained Encoders

A complete listing of URLs for all used pretrained encoders hosted on huggingface.co is provided in Table 8. For monolingual models of each language,

#↓, language→	EN-ES	EN-DE	EN-FI	EN-RU	EN-TR	EN-KO	EN-ZH	EN-JA	EN-TH
muse	112,583	101,931	43,102	48,714	68,611	20,549	39,334	25,969	25,332
wt	1,079,547	1,241,104	338,284	886,760	260,392	319,492	638,900	547,923	107,398

Table 7: Statistics of muse word translations ("muse") and Wikipedia title pairs ("wt").

we made the best effort to select the most popular one (based on download counts).

A.6 Full Table for Comparing with LARGE Models

Table 9 list results across all languages for comparing BASE and LARGE models.

A.7 Future Work

Investigating Other Cross-Lingual Transfer Learning Schemes. We also explored adapting multilingual sentence representation transfer techniques like Reimers and Gurevych (2020) that leverage parallel data. However, we observed no improvement comparing to the main transfer scheme reported in the paper. We plan to investigate existing techniques more comprehensively, and benchmark more results on XL-BEL in the future.

Comparison with in-Domain Parallel Data. While we used general-domain bitexts to cover more resource-poor languages, we are aware that in-domain bitexts exist among several "mainstream" languages (EN, ZH, ES, PT, FR, DE, Bawden et al. 2019).[8] In the future, we plan to also compare with biomedical term/sentence translations on these languages to gain more insights on the impact of domain-shift.

A.8 Number of Model Parameters

All BASE models have ≈110M parameters while LARGE models have ≈340M parameters.

A.9 Hyperparameter Optimisation

Table 10 lists the hyperparameter search space. Note that the chosen hyperparameters yield the overall best performance, but might be suboptimal in any single setting. We used the same random seed across all experiments.

A.10 Software and Hardware Dependencies

All our experiments are implemented using PyTorch 1.7.0 with Automatic Mixed Precision (AMP)[9] turned on. The hardware we use is listed in Table 11. On this machine, the SAP fine-tuning procedure generally takes 5-10 hours with UMLS data. SAP fine-tuning with translation data takes 10 minutes to 5 hours, depending on the amount of the data. Inference generally takes <10 minutes.

[8] http://www.statmt.org/wmt19/biomedical-translation-task.html

[9] https://pytorch.org/docs/stable/amp.html

model	URL
monolingual models	
SAPBERT	https://huggingface.co/cambridgeltl/SapBERT-from-PubMedBERT-fulltext
ESBERT	https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased
DEBERT	https://huggingface.co/dbmdz/bert-base-german-uncased
FIBERT	https://huggingface.co/TurkuNLP/bert-base-finnish-uncased-v1
RUBERT	https://huggingface.co/DeepPavlov/rubert-base-cased
TRBERT	https://huggingface.co/loodos/bert-base-turkish-uncased
KRBERT	https://huggingface.co/snunlp/KR-BERT-char16424
ZHBERT	https://huggingface.co/bert-base-chinese
JABERT	https://huggingface.co/cl-tohoku/bert-base-japanese
THBERT	https://huggingface.co/monsoon-nlp/bert-base-thai
cross-lingual models	
MBERT	https://huggingface.co/bert-base-multilingual-uncased
XLMR	https://huggingface.co/xlm-roberta-base
$\text{XLMR}_{\text{LARGE}}$	https://huggingface.co/xlm-roberta-large
$\text{XLMR}_{\text{LARGE-XNLI}}$	https://huggingface.co/joeddav/xlm-roberta-large-xnli
$\text{XLMR}_{\text{LARGE-SQUAD2}}$	https://huggingface.co/deepset/xlm-roberta-large-squad2

Table 8: A listing of HuggingFace URLs of all pretrained models used in this work.

language→	EN		ES		DE		FI		RU		TR		KO		ZH		JA		TH		**avg**	
model↓	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
SAPBERT	**78.7**	**81.6**	47.3	51.4	22.7	24.7	8.2	10.2	5.8	6.0	26.4	29.7	2.0	2.4	1.9	2.2	3.0	3.2	3.1	3.4	19.9	21.6
$\text{SAPBERT}_{\text{all\_syn}}$	78.3	80.7	55.6	61.3	30.0	34.2	11.8	14.8	9.3	11.3	35.5	39.5	2.0	2.4	6.4	8.2	6.9	8.3	3.0	3.3	23.9	26.4
XLMR	1.0	2.0	0.3	0.7	0.0	0.1	0.1	0.2	0.1	0.2	0.4	0.5	0.0	0.3	0.1	0.2	0.2	0.4	0.0	0.1	0.2	0.5
XLMR + $\text{SAP}_{\text{all\_syn}}$	78.2	81.0	56.4	62.7	31.8	37.3	18.6	22.2	35.4	41.2	42.8	48.9	16.7	21.4	18.8	23.0	24.0	28.1	20.6	27.5	34.3	39.3
$\text{XLMR}_{\text{LARGE}}$	73.0	75.0	20.7	24.6	7.8	9.1	1.9	2.7	3.0	3.3	11.8	13.5	1.2	1.2	0.7	0.9	1.6	1.8	0.9	1.2	12.3	13.3
$\text{XLMR}_{\text{LARGE-XNLI}}$	72.6	75.1	30.1	33.5	10.7	12.2	3.4	4.6	5.9	7.4	16.4	18.4	1.9	2.6	1.3	2.0	2.0	2.5	1.3	2.0	14.6	16.0
$\text{XLMR}_{\text{LARGE-SQUAD2}}$	74.6	76.2	31.4	35.3	11.9	13.2	3.5	4.4	5.2	6.5	16.9	19.2	1.4	1.5	0.6	0.9	1.8	2.1	2.0	2.3	14.9	16.2
$\text{XLMR}_{\text{LARGE}}$ + $\text{SAP}_{\text{all\_syn}}$	78.3	81.3	**61.0**	**66.8**	**35.0**	**40.0**	**25.2**	**29.2**	**41.9**	**47.3**	**46.1**	**52.4**	**22.2**	**26.7**	**23.5**	**29.0**	**28.5**	**33.6**	**28.7**	**35.5**	**39.0**	**44.2**

Table 9: A comparison of BASE (upper half) and LARGE (lower half) multilingual encoders on XL-BEL.

hyperparameters	search space
pretraining learning rate	2e−5
pretraining batch size	512
pretraining training epochs	1
bitext fine-tuning learning rate	2e−5
bitext fine-tuning batch size	{64, 128, 256*}
bitext fine-tuning epochs	{1, 2, 3, 4, 5*, 10}
max_seq_length of tokeniser	25
λ in Online Mining	0.2
α in MS loss (Eq. (1))	2
β in MS loss (Eq. (1))	50
ϵ in MS loss (Eq. (1))	1

Table 10: Hyperparameters along with their search grid. $*$ marks the values used to obtain the reported results. The hparams without any defied search grid are adopted directly from Liu et al. (2021).

hardware	specification
RAM	192 GB
CPU	Intel Xeon W-2255 @3.70GHz, 10-core 20-threads
GPU	NVIDIA GeForce RTX 2080 Ti (11 GB) $\times$ 4

Table 11: Hardware specifications of the used machine. For LARGE model training, we use another server with two NVIDIA GeForce RTX 3090 (24 GB).

A Cluster-based Approach for Improving Isotropy in Contextual Embedding Space

Sara Rajaee[1] and **Mohammad Taher Pilehvar**[1,2]
[1] Iran University of Science and Technology, Tehran, Iran
[2] Tehran Institute for Advanced Studies, Tehran, Iran
sara_rajaee@comp.iust.ac.ir
mp792@cam.ac.uk

Abstract

The representation degeneration problem in Contextual Word Representations (CWRs) hurts the expressiveness of the embedding space by forming an anisotropic cone where even unrelated words have excessively positive correlations. Existing techniques for tackling this issue require a learning process to re-train models with additional objectives and mostly employ a global assessment to study isotropy. Our quantitative analysis over isotropy shows that a local assessment could be more accurate due to the clustered structure of CWRs. Based on this observation, we propose a local cluster-based method to address the degeneration issue in contextual embedding spaces. We show that in clusters including punctuations and stop words, local dominant directions encode structural information, removing which can improve CWRs performance on semantic tasks. Moreover, we find that tense information in verb representations dominates sense semantics. We show that removing dominant directions of verb representations can transform the space to better suit semantic applications. Our experiments demonstrate that the proposed cluster-based method can mitigate the degeneration problem on multiple tasks.[1]

1 Introduction

Despite their outstanding performance, CWRs are known to suffer from the so-called *representation degeneration problem* that makes the embedding space anisotropic (Gao et al., 2019). In an anisotropic embedding space, word vectors are distributed in a narrow cone, in which even unrelated words are deemed to have high cosine similarities. This undesirable property hampers the representativeness of the embedding space and limits the diversity of encoded knowledge (Ethayarajh, 2019).

To better understand the representation degeneration problem in pre-trained models, we analyzed the embedding space of GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019). We found that, despite being extremely anisotropic in all non-input layers from a global sight, the embedding space is significantly more isotropic from a local point of view (when embeddings are clustered and each cluster is made zero-mean). Motivated by this observation and based on previous studies that highlight the clustered structure of CWRs (Reif et al., 2019; Michael et al., 2020), we extend the technique of Mu and Viswanath (2018) with a further clustering step. In our proposal, we cluster embeddings and apply PCA on individual clusters to find the corresponding principal components (PCs) which indicate the dominant directions for each specific cluster. Nulling out these PCs for each cluster renders a more isotropic space. We evaluated our cluster-based method on several tasks, including Semantic Textual Similarity (STS) and Word-in-Context (WiC). Experimental results indicate that our cluster-based method is effective in enhancing the isotropy of different CWRs, reflected by the significant performance improvements in multiple evaluation benchmarks.

In addition, we provide an analysis on the reasons behind the effectiveness of our cluster-based technique. The empirical results show that most clusters contain punctuation tokens, such as periods and commas. The PCs of these clusters encode structural information about context, such as sentence style; hence, removing them can improve CWRs performance on semantic tasks. A similar structure exists in other clusters containing stop words. The other important observation is about verb distribution in the contextual embedding space. Our experiments reveal that verb representations are separated across the tense dimension in distinct

[1]The code for our experiments is available at https://github.com/Sara-Rajaee/clusterbased_isotropy_enhancement/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 575–584
August 1–6, 2021. ©2021 Association for Computational Linguistics

sub-spaces. This brings about an unwanted peculiarity in the semantic space: representations for different senses of a verb tend to be closer to each other in the space than the representations for the same sense that are associated with different tenses of the same verb. Indeed, removing such PCs improves model's ability in downstream tasks with dominant semantic flavor.

2 Isotropy in CWRs

Isotropy is a desirable property of word embedding spaces and arguably any other vector representation of data in general (Huang et al., 2018; Cogswell et al., 2016). From the geometric point of view, a space is called isotropic if the vectors within that space are uniformly distributed in all directions. Lacking isotropy in the embedding space affects not only the optimization procedure (e.g., model's accuracy and convergence time) but also the expressiveness of the embedding space; hence, improving the isotropy of the embedding space can lead to performance improvements (Wang et al., 2020; Ioffe and Szegedy, 2015).

We measure the isotropy of embedding space using the partition function of Arora et al. (2016):

$$F(u) = \sum_{i=1}^{N} e^{u^T w_i} \quad (1)$$

where u is a unit vector, w_i is the corresponding embedding for the i^{th} word in the embedding matrix $\mathrm{W} \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}}$, N is the number of words in the vocabulary, and D is the embedding size. Arora et al. (2016) showed that $F(u)$ can be approximated using a constant for isotropic embedding spaces. Therefore, for the set U, which is the set of eigenvectors of $\mathrm{W}^T\mathrm{W}$, in the following equation, I(W) would be close to one for a perfectly isotropic space (Mu and Viswanath, 2018).

$$\mathrm{I(W)} = \frac{min_{u\in U} F(u)}{max_{u\in U} F(u)} \quad (2)$$

2.1 Analyzing Isotropy in pre-trained CWRs

Using the above metric, we analyzed the representation degeneration problem globally and locally.

Global assessment. We quantified isotropy in all layers for GPT-2, BERT, and RoBERTa on the development set of STS-Benchmark (Cer et al., 2017). Figure 1 shows the trend of isotropy in all layers based on I(W). Clearly, all CWRs are extremely anisotropic in all non-input layers. While

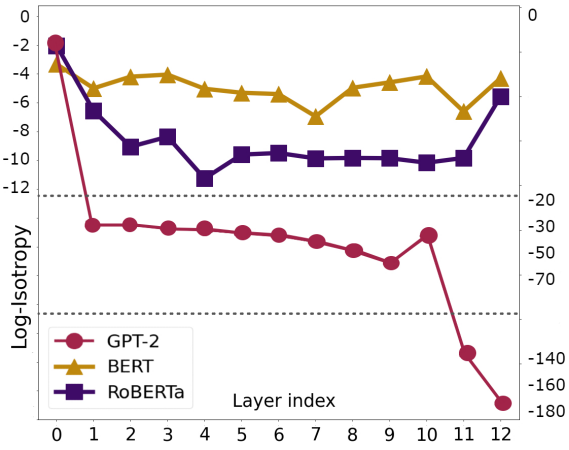


Figure 1: Layer-wise isotropy for different CWRs on the STS-B dev set (↑ log-isotropy: ↑ isotropy). Given the large difference, BERT and RoBERTa are shown on the left axis and GPT-2 on the right.

the isotropy of GPT-2 decreases consistently in upper layers, that for RoBERTa has a semi-convex form in which the last layer (except for the input layer) has the highest isotropy. Also, interestingly, the input layer in GPT-2 is more isotropic than those for the other two models. This observation contradicts with what has been previously reported by Ethayarajh (2019).

Local assessment. In the light of the clustered structure of the embedding space in CWRs (Reif et al., 2019), we carried out a local investigation of isotropy. To this end, we clustered the space using k-means and measured isotropy after making each cluster zero-mean (Mu and Viswanath, 2018). Table 1 shows the results for different number of clusters (each being the average of five runs). When the embedding space is viewed closely, the distribution of CWRs is notably more isotropic. Clustering significantly enhances isotropy for BERT and RoBERTa, making their embedding spaces almost isotropic. However, GPT-2 is still far from being isotropic. This contradicts with the observation of Cai et al. (2021).

A possible explanation for these contradictions is the different metric used by Ethayarajh (2019) and Cai et al. (2021) for measuring isotropy: cosine similarity. Randomly sampled words in an anisotropic embedding space should have high cosine similarities (a near-zero similarity denotes isotropy). However, there are exceptional cases where this might not hold (an anisotropic embedding space where sampled words have near-zero cosine similarities). In Figure 2, we illustrate GPT-

	GPT-2	BERT	RoBERTa
Baseline	5.02E-174	5.05E-05	2.70E-06
$k = 1$	2.49E-220	0.010	0.015
$k = 3$	9.42E-66	0.040	0.290
$k = 6$	**1.40E-41**	0.125	0.453
$k = 9$	1.18E-41	0.131	0.545
$k = 20$	4.06E-47	**0.262**	**0.603**

Table 1: CWRs isotropy after clustering and making each cluster zero-mean separately (results for different number of clusters (k) on STS-B dev set).

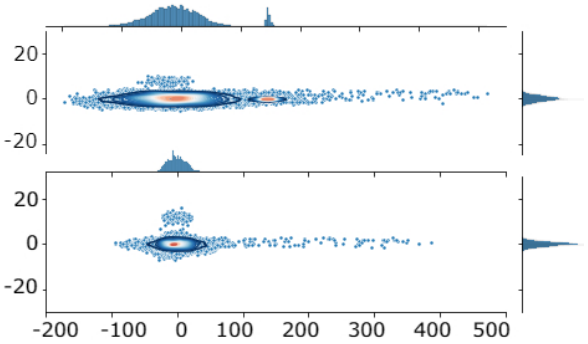


Figure 2: GPT-2 embeddings on STS-B dev set before (top) and after (bottom) a local zero-mean operation.

2 embedding space as an example for such an exceptional cases. Making individual clusters zero-mean (bottom) improves isotropy over the baseline (top). However, the embeddings are still far from being uniformly distributed in all directions. Instead, they are distributed around a horizontal line. This leads to a near-zero cosine similarity for randomly sampled words while the embedding space is anisotropic. Hence, cosine similarity might not be a proper metric for computing isotropy.

3 Cluster-based Isotropy Enhancement

The degeneration problem in the embedding space can be attributed to the training procedure of the underlying models, which are often language models trained through likelihood maximization with the weight tying trick (Gao et al., 2019). Maximizing the likelihood of a specific word embedding (minimizing that for others) requires pushing it towards the direction of the corresponding hidden state, which results in the accumulation of the learnt word embeddings into a narrow cone.

Previous work has shown that nulling out dominant directions of an anisotropic embedding space can make the space isotropic and improve its expressiveness (Mu and Viswanath, 2018). We refer to this as the *global* approach. This method was proposed for static embeddings. Hence, it might not be optimal for contextual embeddings, especially in the light that the latter tends to have a clustered structure. For instance, recent work suggests that word types (e.g., verbs, nouns, punctuations), entities (e.g., personhood, nationalities, and dates), and even word senses (Michael et al., 2020; Loureiro et al., 2021; Reif et al., 2019) create local distinct clustered areas in the contextual embedding space. Moreover, our local assessment shows that it is not necessarily the case that all clusters share the same dominant directions. Hence, discarding dominant directions that are computed globally is not efficient for removing local degenerated directions. Consequently, it is more logical to have a cluster-specific dropping of dominant directions.

Based on these observations, we propose a cluster-based approach for isotropy enhancement. Specifically, instead of determining dominant directions globally, we obtain them separately for different sub-spaces and discard for each cluster only the corresponding cluster-specific dominant directions. To this end, we employ Principal Component Analysis (PCA) to compute local dominant directions in clusters. Geometrically, principal components (PCs) represent those directions in which embeddings have the most variance (maximum elongation). In our proposed method, we first cluster word embeddings using a simple k-means algorithm. After making each cluster zero-mean, the top PCs of every cluster are removed separately. Adding a clustering step helps us to eliminate the local dominant directions of each cluster. We will show in Section 5 that different linguistic knowledge is encoded in the dominant directions of various clusters. Moreover, numerical results show that in comparison with the global approach, our method can make the embedding space more isotropic, even when the fewer number of PCs are nulled out.

4 Experiments

We carried out experiments on the following benchmarks. As for Semantic Textual Similarity (STS), which is the main benchmark for our experiments, we experimented with STS 2012-2016 datasets (Agirre et al., 2012, 2013, 2014, 2015, 2016), the SICK-Relatedness dataset (SICK-R) (Marelli et al., 2014), and the STS benchmark (STS-B). For the STS task, we report results for GPT-2, BERT, and RoBERTa. We also experimented with a number

	Model	STS 2012	STS 2013	STS 2014	STS 2015	STS 2016	SICK-R	STS-B
Baseline	GPT-2	26.49	30.25	35.74	41.25	46.40	45.05	24.8
	BERT-base	42.87	59.21	59.75	62.85	63.74	58.69	47.4
	RoBERTa-base	33.09	56.44	46.76	55.44	60.88	61.28	56.0
Global approach	GPT-2	51.42	69.71	55.91	60.35	62.12	59.22	55.7
	BERT-base	54.62	70.39	60.34	63.73	69.37	63.68	65.5
	RoBERTa-base	51.59	73.57	60.70	66.72	**69.34**	65.82	70.1
Cluster-based approach	GPT-2	**52.40**	**72.71**	**59.23**	**62.19**	**64.26**	**59.51**	**62.3**
	BERT-base	**58.34**	**75.65**	**63.55**	**64.37**	**69.63**	**63.75**	**66.0**
	RoBERTa-base	**54.87**	**76.70**	**64.18**	**67.05**	69.28	**66.93**	**71.4**

Table 2: Spearman correlation performance of three pre-trained models (baseline) on the Semantic Textual Similarity datasets, before and after isotropy enhancement with the global and cluster-based (our) approach.

	RTE	CoLA	SST-2	MRPC	WiC	BoolQ	*Average*
Baseline	54.4	38.0	80.1	70.2	60.0	64.7	61.2
Global approach	56.2	38.8	80.2	72.1	60.7	64.9	62.1
Cluster-based approach	**56.5**	**40.7**	**82.5**	**72.4**	**61.0**	**66.4**	**63.2**

Table 3: Results on the classification tasks (BERT) in terms of accuracy (except for CoLA: Matthew's correlation).

of classification tasks: Recognizing Textual Entailment from the GLUE benchmark (Wang et al., 2018, RTE), the Corpus of Linguistic Acceptability (Warstadt et al., 2019, CoLA), Stanford Sentiment Treebank (Socher et al., 2013, SST-2), Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005, MRPC), Word-in-Context (Pilehvar and Camacho-Collados, 2019, WiC), and BoolQ (Clark et al., 2019). For the classification tasks, we limit our experiments to BERT and extract features to train an MLP. Further details on the datasets and system configuration can be found in Appendix B.

We benchmark our cluster-based approach with the pre-trained CWRs (baseline) and the global method. As it was mentioned before, this method is similar to ours in its elimination of a few top dominant directions but with the difference that these directions are computed globally (in contrast to our local cluster-based computation). The best setting for each model is selected based on performance on the STS-B dev set. The reported results are the average of five runs.

4.1 Results

Tables 2 and 3 report experimental results. As can be seen, globally increasing isotropy can make a significant improvement for all the three pre-trained models. However, our cluster-based approach can achieve notably higher performance compared to the global approach. We attribute this improvement to our cluster-specific discarding of dominant directions. Both global and cluster-based methods null out the optimal number of top dominant directions (tuned separately, cf. Appendix B), but the latter identifies them based on the specific structure of a sub-region in the embedding space (which might not be similar to other sub-regions).

5 Discussion

In this section, we provide a brief explanation for reasons behind the effectiveness of the cluster-based approach through investigating the linguistic knowledge encoded in the dominant local directions. We also show that enhancing isotropy reduces convergence time.

5.1 Linguistic knowledge

Punctuations and stop words. We observed that local dominant directions for the clusters of punctuations and stop words carry structural and syntactic information about the sentences in which they appear. For example, the two sentences "A man is crying." and "A woman is dancing." from STS-B do not have much in common in terms of semantics but are highly similar with respect to their style. To quantitatively analyze the distribution of this type of tokens in CWRs, we designed an experiment based on the dataset created by Ravfogel et al. (2020). The dataset consists of groups in which sentences are structurally and syntactically similar but have no semantic similarity. We picked 200 different structural groups in which each group has six semantically different sentences. Then, using the k-NN algorithm, we calculated the percentage of

Model	Baseline				Removed PCs			
	ST-SM	ST-DM	DT-SM	*Isotropy*	ST-SM	ST-DM	DT-SM	*Isotropy*
GPT-2	48.82	48.19	50.86	2.26E-05	9.32	9.53	9.49	0.17
BERT	13.44	14.24	14.87	2.24E-05	10.31	10.50	10.32	0.32
RoBERTa	5.89	6.31	6.86	1.22E-06	4.78	5.00	4.89	0.73

Table 4: The mean Euclidean distance of a sample occurrence of a verb to all other occurrences of the same verb with the Same-Tense and the Same-Meaning (ST-SM), the Same-Tense but Different-Meaning (ST-DM), and a Different-Tense but the Same-Meaning (DT-SM). Semantically, it is desirable for DT-SM to be lower than ST-DM.

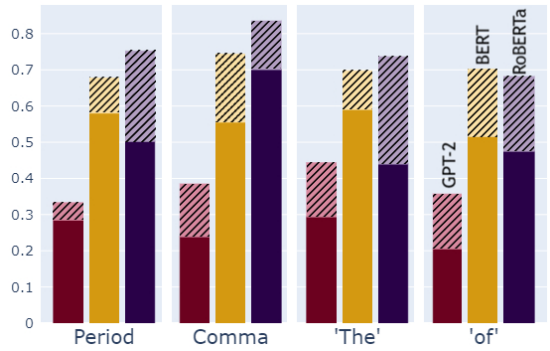


Figure 3: The percentage of nearest neighbours that share similar structural and syntactic knowledge, before (lighter, pattern-filled) and after removing dominant directions in pre-trained CWRs.

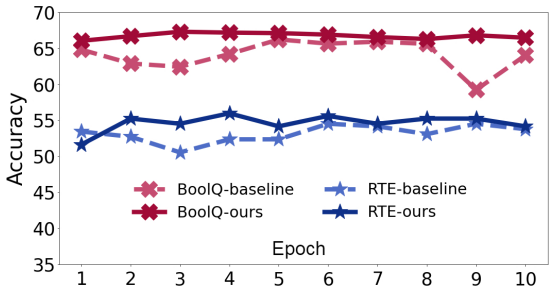


Figure 4: The impact of our cluster-based isotropy enhancement on per-epoch performance for two tasks.

nearest neighbours which are in the same group before and after removing local dominant directions. We evaluated this for period and comma, which are the most frequent punctuations, and "the" and "of" as the most contextualized stop words (Ethayarajh, 2019). The reported results in Figure 3 show that the representations for punctuations and stop words are biased toward structural and syntactic information of sentences; hence, removing their dominant directions reduces the number of same-group nearest neighbours. The improvement from our local isotropy enhancement can be partially attributed to attenuating this type of bias.

Verb Tense. Our experiments show that tense is more dominant in verb representations than sense-level semantic information. To have a precise examination of this hypothesis, we used SemCor (Miller et al., 1993), a dataset comprising around 37K sense-annotated sentences. We collected representations for polysemous verbs that had at least two senses occurring a minimum of 10 times. Then, for each individual verb, we calculated Euclidean distance to the contextual representation of the same verb: (1) with the same tense and the same meaning, (2) with the same tense but a different meaning, and (3) with a different tense and the same meaning. The experimental results reported in Table 4 confirm the hypothesis and show the effectiveness of the cluster-based approach in bringing together verb representations that correspond to the same sense, even if they have different tense.

5.2 Convergence time

In the previous experiments, we showed that the contextual embeddings are extremely anisotropic and highly correlated. Such embeddings can slow down the learning process of deep neural networks. Figure 4 shows the trend of convergence for the BoolQ and RTE tasks (dev sets). By decreasing the correlation between embeddings, our method can reduce convergence time.

6 Conclusions

In this paper, we proposed a cluster-based method to address the representation degeneration problem in CWRs. We empirically analyzed the effect of clustering and showed that, from a local sight, most clusters are biased toward structural information. Moreover, we found that verb representations are distributed based on their tense in distinct sub-spaces. We evaluated our method on different semantic tasks, demonstrating its effectiveness in removing local dominant directions and improving performance. As future work, we plan to study the effect of fine-tuning on isotropy and on the encoded linguistic knowledge in local regions.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Cogswell, Faruk Ahmed, Ross B. Girshick, Larry Zitnick, and Dhruv Batra. 2016. Reducing overfitting in deep networks by decorrelating representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *CoRR*, abs/1907.12009.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. In *Advances in Neural Information Processing Systems*, volume 31, pages 1334–1345. Curran Associates, Inc.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. 2018. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and Evaluation of Language Models for Word Sense Disambiguation. *Computational Linguistics*, pages 1–57.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Julian Michael, Jan A. Botha, and Ian Tenney. 2020. Asking without telling: Exploring latent ontologies in contextual representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6792–6812, Online. Association for Computational Linguistics.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shauli Ravfogel, Yanai Elazar, Jacob Goldberger, and Yoav Goldberg. 2020. Unsupervised distillation of syntactic information from contextualized word representations. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 91–106, Online. Association for Computational Linguistics.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, volume 32, pages 8594–8603. Curran Associates, Inc.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

A Isotropy statistics

Table 5 shows isotropy statistics for GPT-2, BERT, and RoBERTa. GPT-2's embedding space is extremely anisotropic in upper layers. Hence, more PCs are required to be eliminated to make this embedding space isotropic in comparison to BERT and RoBERTa, both in the cluster-based approach and the global one (Mu and Viswanath, 2018). Also, in almost all layers, BERT has higher a isotropy than RoBERTa.

Model	GPT-2	BERT	RoBERTa
layer 0	1.5E-02	4.6E-04	9.1E-03
layer 1	9.9E-24	9.9E-06	2.7E-07
layer 2	**2.8E-23**	6.3E-05	8.7E-10
layer 3	6.1E-26	8.8E-05	4.2E-09
layer 4	1.6E-27	9.2E-06	5.4E-12
layer 5	3.0E-30	4.8E-06	2.4E-10
layer 6	1.6E-32	3.9E-06	3.1Ef-10
layer 7	1.3E-37	1.1E-07	1.3E-10
layer 8	3.4E-45	1.0E-05	1.4E-10
layer 9	6.4E-55	2.5E-05	1.3E-10
layer 10	4.1E-32	6.9E-05	6.7E-11
layer 11	1.8E-132	2.4E-07	1.4E-10
layer 12	5.0E-174	**5.0E-05**	**2.7E-06**

Table 5: Per-layer isotropy on the STS-B dev set. Numbers have been calculated based on I(W).

B Experimental Setup

B.1 Dataset details

STS. In the Semantic Textual Similarity task, the provided labels are between 0 and 5 for each paired sentence. We first calculate sentence embeddings by averaging all word representations in each sentence and then compute the cosine similarity between two sentence representations as a score of semantic relatedness of the pair.

RTE. The Recognizing Textual Entailment dataset is a classification task from the GLUE benchmark (Wang et al., 2018). Paired sentences are collected from different textual entailment challenges and labeled as *entailment* and *not-entailment*.

CoLA. The Corpus of Linguistic Acceptability (Warstadt et al., 2019) is a binary classification task in which sentences are labeled whether they are grammatically acceptable.

SST-2. The Stanford Sentiment Treebank (Socher et al., 2013) is a binary sentiment classification task.

MRPC. The Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) consists of paired sentences, and the goal is determining whether, in a pair, sentences share similar semantics or not.

WiC. Word-in-Context (Pilehvar and Camacho-Collados, 2019) is a binary classification task in which it should be determined if a target word in two different contexts refers to the same meaning.

BoolQ. Boolean Questions (Clark et al., 2019) is a Question Answering classification task. Every sample includes a passage and a yes/no question about the passage.

B.2 Configurations

For the classification tasks, we trained a simple MLP on the features extracted from BERT. The proposed cluster-based approach has two hyperparameters: the number of clusters and the number of PCs to be removed. We selected both of them from range [5, 30] and tuned them on the STS-B dev set. In the cluster-based approach,The optimal number of clusters for GPT-2, BERT, and RoBERTa are respectively 10, 27, and 27. For BERT and RoBERTa, 12 top dominant directions have been removed, while the number is 30 for GPT-2 regarding its extremely anisotropic embedding space. The tuning of the number of PCs to be eliminated in the global method has been done similarly to the cluster-based approach (on the STS-B dev set): 30, 15, and 25 for GPT-2, BERT, and RoBERTa, respectively.

C Isotropy on STS datasets

In Table 6, we present the isotropy of the contextual embedding spaces calculated using I(W) on the STS benchmark. The results reveal the effectiveness of the proposed method in enhancing the isotropy of the embedding space.

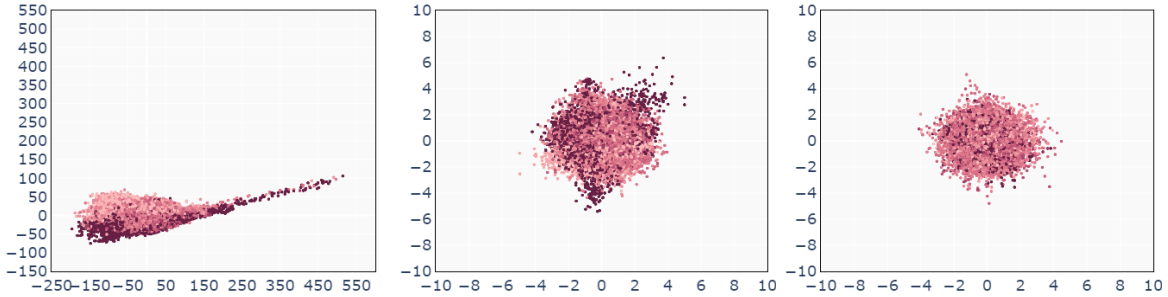
D Word frequency bias in CWRs

CWRs are biased towards their frequency information, and words with similar frequency create local regions in the embedding space (Gong et al., 2018; Li et al., 2020). From the semantic point of view, this is certainly undesirable given that words with similar meanings but different frequencies could be

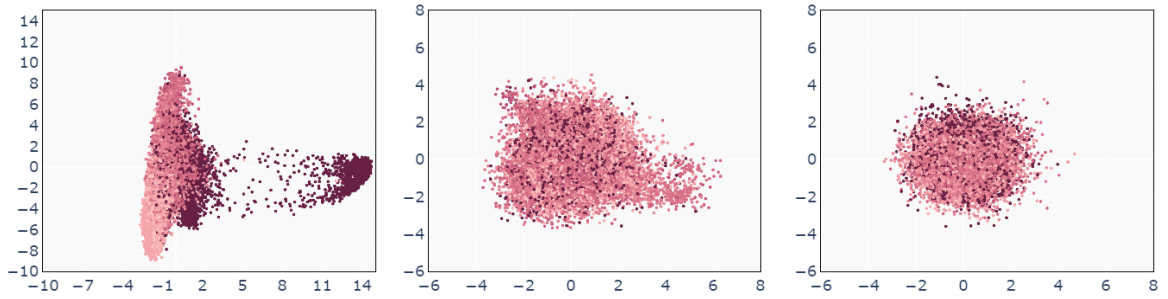
Model	STS 2012	STS 2013	STS 2014	STS 2015	STS 2016	SICK-R	STS-B
				Baseline			
GPT-2	1.4E-178	1.0E-170	1.4E-172	2.9E-177	6.0E-174	9.9E-140	2.6E-105
BERT	3.1E-05	1.9E-04	2.6E-04	3.7E-07	2.8E-04	4.2E-05	1.1E-04
RoBERTa	3.1E-06	3.1E-07	3.8E-06	3.8E-06	3.5E-06	3.7E-07	2.9E-06
				Global approach			
GPT-2	0.57	0.40	0.05	0.12	0.60	0.57	0.51
BERT	0.48	0.41	0.55	0.72	0.65	0.63	0.58
RoBERTa	0.67	0.87	0.87	0.84	0.85	0.90	0.88
				Cluster-based approach			
GPT-2	**0.71**	**0.74**	**0.47**	**0.74**	**0.74**	**0.78**	**0.70**
BERT	**0.68**	**0.61**	**0.77**	**0.81**	**0.75**	**0.82**	**0.73**
RoBERTa	**0.89**	**0.91**	**0.93**	**0.92**	**0.89**	**0.94**	**0.90**

Table 6: Isotropy of CWRs on multiple STS datasets calculated based on I(W); a higher value indicates a more isotropic embedding space. Our cluster-based method significantly increases the isotropy of embedding space on all datasets.

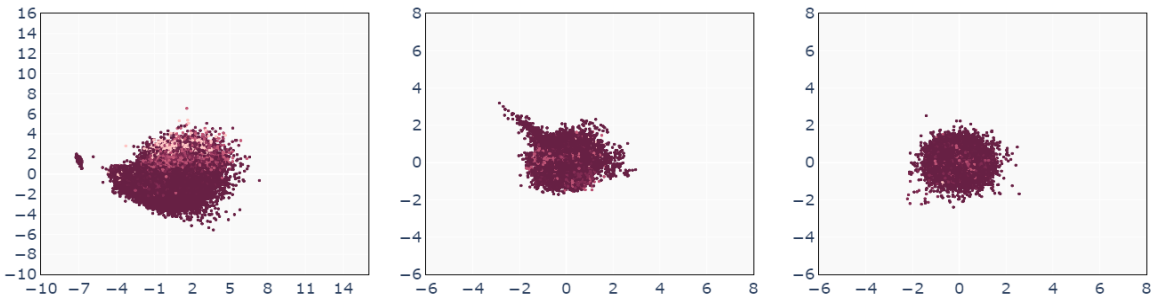
located far from each other in the embedding space. This phenomenon can be seen in Figure 5. The encoded knowledge in the local dominant directions partly correspond to frequency information. The embedding space visualization reveals that our approach performs a decent job in removing frequency bias in pre-trained models.



(a) GPT-2 - Baseline (b) GPT-2 - Global approach (c) GPT-2 - Cluster-based approach



(d) BERT - Baseline (e) BERT - Global approach (f) BERT - Cluster-based approach



(g) RoBERTa - Baseline (h) RoBERTa - Global approach (i) RoBERTa - Cluster-based approach

Figure 5: Contextual Word Representations visualization using PCA on STS-B dev set. Colors indicate word frequency in the Wikipedia dump (the lighter point, the more frequent).

Unsupervised Enrichment of Persona-grounded Dialog with Background Stories

Bodhisattwa Prasad Majumder[♣] **Taylor Berg-Kirkpatrick**[♣]
Julian McAuley[♣] **Harsh Jhamtani**[♢]
[♣]Department of Computer Science and Engineering, UC San Diego
{bmajumde, tberg, jmcauley}@eng.ucsd.edu
[♢]School of Computer Science, Carnegie Mellon University
jharsh@cs.cmu.edu

Abstract

Humans often refer to personal narratives, life experiences, and events to make a conversation more engaging and rich. While persona-grounded dialog models are able to generate responses that follow a given persona, they often miss out on stating detailed experiences or events related to a persona, often leaving conversations shallow and dull. In this work, we equip dialog models with 'background stories' related to a persona by leveraging fictional narratives from existing story datasets (e.g. ROCStories). Since current dialog datasets do not contain such narratives as responses, we perform an unsupervised adaptation of a retrieved story for generating a dialog response using a gradient-based rewriting technique. Our proposed method encourages the generated response to be *fluent* (i.e., highly likely) with the dialog history, *minimally different* from the retrieved story to preserve event ordering and *consistent* with the original persona. We demonstrate that our method can generate responses that are more diverse, and are rated more engaging and human-like by human evaluators, compared to outputs from existing dialog models.

1 Introduction

Humans often rely on specific incidents and experiences while conversing in social contexts (Dunbar et al., 1997). Responses from existing chitchat dialog agents often lack such specific details. To mitigate this, some prior work has looked into assigning personas to dialog agents (Zhang et al., 2018; Majumder et al., 2020). However, persona descriptions are often shallow and limited in scope, and while they lead to improvements response specificity, they still lack the level of detail with which humans share experiences.

In this work, we propose methods to enrich dialog personas with relevant background events using fictional narratives from existing story datasets such as ROCStories (Mostafazadeh et al., 2016). For example, for a persona attribute 'I have two children and a dog,' we are able to identify a relevant narrative from a story corpus (Figure 1). However, such stories may not directly fit fluently in the dialog context. Thus, retrieved stories should be adapted to construct a response that is fluent and relevant to the context. Since existing datasets (such as PersonaChat (Zhang et al., 2018)) do not contain responses with such background stories, such adaptation has to be done in an unsupervised fashion with decoders trained to generate responses conditioned only on a dialog history and persona.

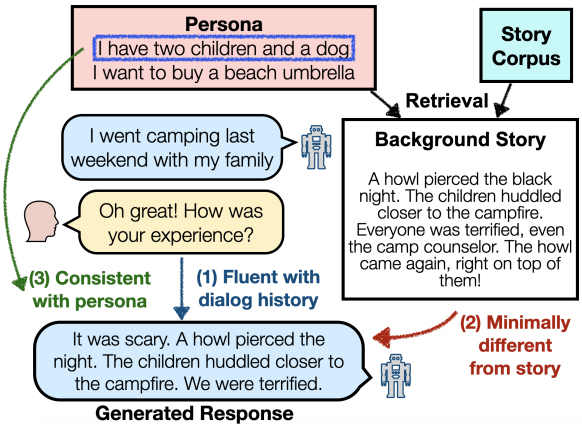


Figure 1: We enrich agent personas with 'background stories' from an existing corpus. We propose a gradient-based technique which encourages the generated response to be fluent with the dialog history, minimally different from the retrieved story, and consistent with the persona. The proposed approach leads to more specific and interesting responses.

To adapt a retrieved narrative incident as a relevant background story, we use a decoding procedure which encourages the generated response to (1) be fluent with the dialog history, (2) be consistent with the original persona, and (3) be minimally different from the retrieved story. While fluency with dialog context is encouraged directly by the likelihood as per the underlying language model

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 585–592
August 1–6, 2021. ©2021 Association for Computational Linguistics

the remaining two constraints are incorporated via iterative updates to the decoder output distributions at inference time. Our inference-time decoding method is different from the only recent effort by Su et al. (2020) that leverages non-dialog data (forum comments, book snippets) as distant labels to train dialog systems with supervision. Our contributions can be summarized as follows:

- We propose a novel approach to enrich dialog agent personas with relevant backstories, relying only on existing story datasets.
- We propose to use an unsupervised back-propagation based decoding procedure[1] to adapt the relevant stories such that the resulting response is fluent with the dialog history and consistent with the dialog agent persona. Our method works with a model trained just with dialog data i.e. without access to story corpus at training time.
- Our experiments demonstrate that the proposed approach results in much more engaging and specific dialog outputs in a persona-grounded dialog setup. This fills a gap in existing dialog models which often lack the capability to generate responses about specific events and experiences relevant to persona attributes.

2 Unsupervised Persona Enrichment with Background Stories

Given dialog history h and persona C consisting of several (typically 3-5, example shown in Figure 1) attributes, our goal is to construct a dialog response x. Our underlying model is based on the discrete persona attribute choice model from Majumder et al. (2020). To generate a dialog utterance x, we first sample a persona attribute $c \sim p(c|h)$ conditioned on the dialog history h. x is then generated conditioned on the dialog history and the chosen persona attribute. The underlying dialog model's decoder is initialized with a pretrained GPT-2 model, and is fine-tuned on the PersonaChat dataset (Zhang et al., 2018). However, in our current setup, we also have to identify relevant background stories and use them to construct fluent responses at decoding time. Therefore, we propose a different decoding procedure.

To generate a response, we first sample a persona attribute $c \sim p(c|h)$. Next we retrieve stories corresponding to the persona attribute c (Section 2.1). However, the underlying dialog model is trained to generate responses conditioned only on the dialog history and persona. To incorporate the retrieved story in the response, we perform gradient-based inference (Section 2.2), that only assumes a left-to-right language model trained on dialog context and responses, and the story is handled at decoding time in an unsupervised fashion. We refer to the proposed method as **PABST** (Unsupervised **P**erson**A** enrichment with **B**ackground **ST**ories).

2.1 Retrieving Relevant Stories

For a persona attribute c, we aim to identify relevant stories from a story corpus. Toward this goal, we rank the stories using the F1 component of BERT-score (Zhang et al., 2020) based retrieval using the persona attribute c as the query and the highest scoring story is chosen. Note that many of the stories are written in the third person. For use as background stories, we must first transform them to first–person. Following prior work (Brahman and Chaturvedi, 2020), we identify the protagonist of such stories as the most frequently occurring character. Thereafter, we use co-reference resolution (Lee et al., 2017) to identify all words or phrases that refer to the protagonist. Finally, all words or phrases so identified are replaced with suitable first person pronouns (e.g. 'his books' to 'my books').

2.2 Gradient-based Inference

Our underlying dialog model is not trained to condition on a retrieved story, and cannot be directly used to construct a desirable response using s. To tackle this, we consider a decoding strategy which, in addition to fluency with history h, encourages response x to follow two soft constraints: (1) be minimally different from story s, and (2) be consistent with persona c.

First, we generate an initial response based only on the dialog history. Then we perform an iterative procedure which alternates between performing a forward pass on the language model to encourage fluency, and a backward pass which updates the response via back-propagation to respect the two soft constraints. However, x is discrete, and cannot be directly updated using gradients from back-propagation. Instead, we maintain and update a soft representation o of x, where o_i corresponds to the last hidden state representation for the i^{th} token position, i.e., $p(x_i) \sim \text{softmax}(Wo_i/\tau)$, where τ is the temperature parameter, W is the embedding

[1] Code can be found at https://github.com/majumderb/pabst

matrix, and $Wo_i \in \mathcal{R}^V$ (V is the vocabulary size). Our approach is inspired by recent works that use gradient-based decoding for text generation with soft constraints (Dathathri et al., 2020; Qin et al., 2020). Next we describe the backward and forward passes of the iterative procedure.

Backward Pass with Soft Constraints We define the following soft constraints on response x:
(1) **Divergence from story:** We want to encourage x to be *minimally different* from the story s. Following prior work (Qin et al., 2020), we compute a cross entropy loss (denoted by cross-entr henceforth) with story $s = \{s_1, \ldots, s_T\}$ tokens as labels and $Wo_1, \ldots, Wo_T$ as the logits.
(2) **Consistency to persona:** We want x to be *consistent with persona attribute* c. Consider a classifier $q_\phi(o, c)$ which predicts the probability of x (or rather the soft representation o of x) entailing c. The classifier $q_\phi(o, c)$ is a bag-of-words classification head on decoder hidden states o, fine-tuned on the Dialogue-NLI dataset (Welleck et al., 2019) to predict whether pairs of persona attributes and responses are entailed or not. The objective to maximize can be written as:

$$\mathcal{L}(c, s; o) = \lambda_c \log q_\phi(o, c) - \lambda_d \,\text{cross-entr}(s, Wo)$$

where λ_c and λ_d are hyper-parameters. We update o through back-propagation by computing the gradient $\nabla_o \mathcal{L}(c, s; o)$, while keeping the model parameters constant. Let the resulting o after the gradient-based updates be denoted by o^b.

Forward Pass to Encourage Fluency Next we perform a forward pass of the underlying dialog model, with the goal of regularizing the hidden states towards the unmodified language model values. On computing the forward pass at the j^{th} token, we mix the final hidden states o_j^f from the forward pass with o_j^b computed in the backward pass, via weighted addition to get the resulting $o_j = \gamma \times o_j^f + (1 - \gamma) \times o_j^b$, where $\gamma \in (0, 1)$ is a hyperparameter. The resulting o_j is used for computing the logits at the next time step $j + 1$.

We initialize the output response by performing greedy decoding from the underlying dialog model, conditioned on the dialog history and persona attribute. Then we iteratively update o by alternate backward and forward passes. We sample the final response $x \sim \text{softmax}(Wo/\tau)$. In practice, we found that 5 iterations are sufficient to generate good quality outputs.

Method	Training	Decoding	D-1	D-2	ENTR
W/o Story Data					
TRANSFERO	PERSONA	Nucleus	0.05	0.11	1.21
DISCCHOICE	PERSONA	Nucleus	0.15	0.25	1.25
DISCCHOICE	CS-KB	Nucleus	0.87	1.07	2.04
With Story Data					
DISCCHOICE	PSEUDO	Nucleus	0.91	2.45	2.89
DISCCHOICE	MULTITASK	Nucleus	0.99	2.54	2.71
DISCCHOICE	PERSONA	RETRIEVAL	2.56	9.67	3.86
PABST (Ours)	PERSONA	Grad. Inf.	1.56	3.57	3.21

Table 1: Diversity metrics on the PersonaChat test set. D-1/2 is the % of distinct uni- and bi-grams. ENTR is the geometric mean of n-gram entropy. Grad. Inf. is the unsupervised gradient-based decoding as opposed to Nucleus sampling (Holtzman et al., 2020).

3 Experiments

We evaluate methods in terms of their capability to generate diverse, fluent and engaging responses. Hyperparameters are noted in Appendix §A.

Datasets We experiment with the PersonaChat dialog dataset (Zhang et al., 2018) consisting of 131,438 utterances for training, 15,602 for validation, and 15,024 for testing. For stories, we use the training split of the ROCStories dataset (Mostafazadeh et al., 2016), that consists of 78,529 stories, each typically of 4 to 5 sentences.

Baselines We consider two broad groups of models as baselines: **(1)** ***Without access to story corpus***: We use finetuned GPT2 (**TRANSFERO**) on PersonaChat, and the discrete persona attribute choice model (**DISCCHOICE**) from Majumder et al. (2020). We also consider a version of DISCCHOICE which enriches personas with inferences from a commonsense knowledge base (**CS-KB**). **(2)** ***Baselines using story corpus*:** To allow DISCCHOICE models to generate story-like responses, we adapt an alternative training regime (**PSEUDO**) from (Su et al., 2020), where we randomly replace some of the target dialog responses with retrieved stories—treating them as pseudo labels. Finally, we also consider a **MULTITASK** training setup from (Su et al., 2020), wherein the decoder is trained on PersonaChat as well as with a language modeling objective on ROCStories. We additionally consider a **RETRIEVAL** baseline that uses the retrieved story verbatim as the dialog response.

3.1 Automatic Evaluation

We hypothesize that that the proposed approach to leverage external non-dialog data can increase the diversity of the generated responses. Following

PABST vs.	TRANSFERO		DISCCHOICE		RETRIEVAL		PSEUDO		MULTITASK		w/o DNLI		Gold	
Aspect	win	loss	win	loss	win	loss	win	loss	win	loss	win	loss	win	loss
Sensible	**83.8**	5.6	**71.4**	10.1	**72.7**	15.7	**94.3**	3.1	**92.8**	1.5	**81.7**	6.8	29.1	**33.6**
Engaging	**96.7**	1.8	**82.6**	6.7	**87.2**	5.3	**94.7**	2.1	**94.1**	2.6	**70.9**	14.8	**70.2**	15.6

Table 2: Pairwise comparison (% win/loss cases, ties not reported) between responses from PABST and from other baselines as well as gold. All differences are significant with $p < 0.05$. Cohen's Kappa (Cohen, 1960) for sensibility and engagement were 0.79 and 0.82 respectively. Complete details of the human evaluations are in Appendix §B.

Dialog History Agent: How was your last trip? User: It was great. I went to Utah and saw the Canyonlands.	**Dialog History** Agent: I work as stunt double. What about you? User: I'm an engineer. I just got back from work. How's yours?
Retrieved Story: I went on a trip. I went to Florida. When we got there, there was no air conditioning. I was irritated. It was too hot.	**Retrieved Story:** I was a working actor. For my new role I was a stunt double. I started working on new stunts. When I jumped off a building yesterday I got hurt. I had to be rushed to the hospital.
DISCCHOICE: That's great. You must be adventurous.	**DISCCHOICE:** That's great. I am looking for a job.
RETRIEVAL: I went on a trip. I went to Florida. When we got there, there was no air conditioning. I was irritated. It was too hot.	**RETRIEVAL:** I was a working actor. For my new role I was a stunt double. I started working on new stunts. When I jumped off a building yesterday I got hurt. I had to be rushed to the hospital.
PSEUDO: I have a house in Florida.	**PSEUDO:** I got a new job. I am an actor.
MULTITASK: I like Florida. I went to the beaches.	**MULTITASK:** I started a new role. I got hurt.
PABST ($\lambda_d = 5$): I went on a trip. I went to Florida. I have a house there. We do not have air condition. It was hot.	**PABST ($\lambda_d = 5$):** I was working as a stunt double. I jumped off a building. I got hurt. I had to be rushed to the hospital.
PABST ($\lambda_d = 1$): That's great. I frequently go to Florida. I have a house there but it gets hot in summer.	**PABST ($\lambda_d = 1$):** Yesterday while I was working, I jumped off a building and I got hurt. I had to be taken to the hospital.

Table 3: Generations from different models. More examples are in Appendix §C.

prior work (Li et al., 2016), we report the percentage of distinct uni-grams and bi-grams (**D-1** and **D-2** respectively). Note that these values do not capture the actual frequency distribution of different word types. Therefore, we also report the geometric mean of entropy values of empirical frequency distributions of n-grams of words (n $\in \{1, 2, 3\}$) (Jhamtani et al., 2018), denoted by **ENTR**.

We observe that methods that use story data show much higher diversity compared to methods that do not (Table 1). Among methods using story data, gradient-based decoding (PABST) performs better than DISCCHOICE trained with PSEUDO or MULTITASK. Note that just using RETRIEVAL outputs as-is leads to even more diverse outputs than PABST. However, they are much less sensible with the context, as shown in human evaluations.

3.2 Human Evaluation

Since we do not have ground truth story-like responses in the dialog dataset, we perform human evaluation with 150 test examples to investigate if PABST generates responses that are 1) **sensible** with the dialog history and 2) **engaging**. We hired two Anglophone (Lifetime HIT acceptance % > 85) annotators for every test sample. The order of the systems present in the interface is randomized. A snapshot of the human evaluation interface is provided in Appendix §C. All differences in values from human evaluations are significant with $p < 0.05$ from bootstrap tests on 1000 subsets of size 50. Cohen's Kappa (Cohen, 1960) to measure inter-annotator agreement for sensibility and engagement were 0.79 and 0.82 respectively.

From the results (shown in Table 3), we note that in comparison to responses from baselines, responses from PABST are more engaging and more sensible with respect to the dialog history. We further make following observations. Firstly, using the gradient-based decoding approach with retrieved stories (PABST) works significantly better than using distant supervision with stories data (PSEUDO and MULTITASK). Secondly, background stories provide sufficient detail for an engaging conversation compared to DISCCHOICE which expands persona attributes using commonsense knowledge (Majumder et al., 2020). Finally, we also observe that PABST performs worse when we do not use the consistency constraint (w/o DNLI).

Choice of λ_d We also experiment with different values of the weight for the divergence term (λ_d) in $\mathcal{L}$: High ($\lambda_d = 5$), Moderate ($\lambda_d = 1$),

and Low ($\lambda_d = 0.05$). We consider 100 samples for this experiment. We attribute a high λ_d to responses strictly copying the story. We find that PABST (moderate λ_d) wins wins 81.2% and 69.1% cases against PABST (high λ_d) on 'sensible' and 'engaging' response criteria respectively. Similarly, PABST (moderate λ_d) wins 93.2% and 84.7% cases against PABST (low λ_d) in terms of sensibility and engagement respectively.

Qualitative Analysis Table 3 shows responses generated by different baselines. We observe that PABST is able to follow the retrieved story (same as output from RETRIEVAL) while modifying the response to be conversation-like and sensible with dialog history. Responses from other baselines remain verbose or incoherent. Mirroring the human evaluation, we observe that choosing a higher λ_d makes the model to almost repeat the retrieved story but a lower value smooths the output to make it more sensible with the ongoing dialog.

4 Related Work

A desired impact of the proposed approach is increase in diversity of the generated responses. To tackle the issue of diversity in dialog model outputs, prior work has focused on decoding strategies such as diversity-promoting sampling (Holtzman et al., 2020); training strategies such as discouraging undesirable responses via unlikelihood training (Li et al., 2020); model changes such as using stochastic variables (Serban et al., 2017); and using external data such as forum data (Su et al., 2020) or external knowledge bases (Majumder et al., 2020). In contrast to these, our proposed method generates responses with background stories using a gradient-based decoding approach.

One of the steps in our proposed approach is to retrieve relevant stories from an external corpus. Prior work has explored using retrieval of similar dialog instances as an initial step in improving response diversity and other human-like desiderata in dialog (Roller et al., 2020; Weston et al., 2018). Distant supervision by using retrieved text snippets as pseudo responses has been explored in prior work (Su et al., 2020; Roller et al., 2020). We use an external data source to improve dialog responses, a theme shared with some efforts in other tasks such as machine translation (Khandelwal et al.). The use of narrative text in dialog has been explored in prior work, mostly as a 'script' or template for conversation (Xu et al., 2020; Zhu et al., 2020). We adapted a BERT-based retrieval method (Zhang et al., 2020) in our case to retrieve relevant story given dialog context and use retrieved story in the decoding phase.

Gradient-based for text generation with soft constraints has been explored in prior work (Dathathri et al., 2020; Qin et al., 2020). Song et al. (2020) focused on generating response which are consistent to given persona. Differently, we use a gradient-based decoding to generate a dialog response while honoring constraints such as consistency to persona and similarity to retrieved story.

5 Conclusion

We propose a method to enrich persona-grounded dialog with background stories at the inference time only using an existing corpus of non-conversational narratives—opening up new ways to generate enriched and engaging responses. One of the limitations of PABST is the assumption of the background story at every turn. As future work, we can include a decision step to decide if we need to incorporate a background story or not, given the dialog history. We can further explore ways to use retrieved stories over multiple turns instead of a single turn.

Acknowledgements

We thank anonymous reviewers for providing valuable feedback. BPM is partly supported by a Qualcomm Innovation Fellowship and NSF Award #1750063. Findings and observations are of the authors only and do not necessarily reflect the views of the funding agencies.

Impact Statement

In this work, we discuss ways to make a dialog system to generate more engaging responses. Since we use a finetuned version of a pretrained generative model, we inherit the general risk of generating biased or toxic language, which should be carefully filtered. Furthermore, the generations may incorporate biases that are already present in the dialog dataset and story dataset due to crowd-sourced data collection. Hence, we cautiously advise any developer who wishes to use a different story dataset for the background stories to be aware of the biases present in the dataset. Finally, we also note that experiments in this paper are limited only to English language.

References

Faeze Brahman and Snigdha Chaturvedi. 2020. Modeling protagonist emotions for emotion-aware storytelling. In *EMNLP*, pages 5277–5294.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *ICLR*.

Robin IM Dunbar, Anna Marriott, and Neil DC Duncan. 1997. Human conversational behavior. *Human nature*, 8(3):231–246.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Learning to generate move-by-move commentary for chess games from large-scale social forum data. In *ACL 2018*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. *CoRR*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT*.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *ACL*.

Bodhisattwa Prasad Majumder, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Julian J. McAuley. 2020. Like hiking? you probably enjoy nature: Persona-grounded dialog with commonsense expansions. In *EMNLP*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *CoRR*, abs/1604.01696.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *EMNLP*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*.

Haoyu Song, Wei-Nan Zhang, Jingwen Hu, and Ting Liu. 2020. Generating persona consistent dialogues by exploiting natural language inference. In *AAAI*.

Hui Su, Xiaoyu Shen, Sanqiang Zhao, Xiao Zhou, Pengwei Hu, Randy Zhong, Cheng Niu, and Jie Zhou. 2020. Diversifying dialogue generation with non-conversational text. In *ACL*.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *ACL*.

Jason Weston, Emily Dinan, and Alexander H. Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *SCAI@EMNLP*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2020. Enhancing dialog coherence with event graph grounded content planning. In *IJCAI*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *ICLR*.

Yutao Zhu, Ruihua Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. 2020. Scriptwriter: Narrative-guided script generation. In *ACL*.

A Implementation Details

We obtain the PersonaChat dataset from ParlAI repository[2]. ROCStories dataset is obtained from the repository of original release[3]. We adapted codes from original PPLM (Dathathri et al., 2020) repository[4] and modified them for our own objective function.

Network architecture For the generator network, we use GPT2 (Transformer with 12 layers, 768 hidden size, 12 heads— `gpt2-small`[5]) following the state-of-the-art model (Wolf et al., 2019) from Conv-AI2 competition. The decoder has total of 124 Million parameters. We used the pre-trained decoder model obtained from (Majumder et al., 2020).

Hyperparameters PABST does not require any training since we perform gradient-based decoding at the inference time. For our best method, in objective function $\mathcal{L}$, we use λ_d as 1 and λ_c as 1. We keep generation length to be 100 to encourage longer generations. We train the consistency classifier using code from PPLM repository[6]. The weight γ for mixing forward and backward passes was set to 0.45. For PSEUDO, we substitute a random 30% of the original target responses with retrieved stories.

B Human Evaluations Setup

Figure 2 shows a sample question for the pairwise comparison between response generated by PABST and a baseline for sensibility and engagingness. A similar setup was used to measure performance between PABST variants with different λ_d values $(0.5, 1, 5)$.

C Qualitative Examples

Tables 4 and 5 show generations from PABST and baselines for sampled text examples.

[2] http://parl.ai/downloads/personachat/personachat.tgz

[3] https://www.cs.rochester.edu/nlp/rocstories/

[4] https://github.com/uber-research/PPLM

[5] https://github.com/huggingface/transfer-learning-conv-ai

[6] https://github.com/uber-research/PPLM/blob/master/run_pplm_discrim_train.py

Dialog History Agent: I volunteer in shelters. User: It is a noble act.
DISCCHOICE: Thank you. I like helping people.
RETRIEVAL: I was at my house bored. I couldn't figure out what to do. I looked on TV for answers. I saw a commercial about homeless people. So I decided to volunteer at a homeless shelter.
PSEUDO: Last week, I went to the local animal shelter on the pet adaptation day.
MULTITASK: Thank you! What about you?
PABST ($\lambda_d = 5.0$)**:** I was bored. I wanted to help people. I looked on TV about homeless people. So I decided to volunteer at a homeless shelter.
PABST ($\lambda_d = 1.0$)**:** Thank you. I was inspired by a TV commercial and I volunteered at the homeless shelter.

Table 4: Generations from different models

Dialog History Agent: I enjoy American sports. What sports do you like? User: I love playing volleyball. I was in the university team.
DISCCHOICE: That's great. I like adventure sports.
RETRIEVAL: I always thought golf was for old people. One day, my dad invited me to play golf with him. I reluctantly accepted his invitation. On the golf range, I saw lots of people of my age. I realized golf was a sport for everyone to enjoy regardless of age.
PSEUDO: Last week, I went to play golf. My dad invited me. I was reluctant.
MULTITASK: That's great! I play volleyball during weekend.
PABST ($\lambda_d = 5.0$)**:** I like playing golf. I always thought it is for old people. Recently I went to a golf range. But I saw lots of people of my age. I realized golf is a sports for everyone.
PABST ($\lambda_d = 1.0$)**:** That's great. I like playing golf. I always thought it is for old people. Recently I went to a golf range. But I saw lots of people of my age. I realized golf is a sports for everyone.

Table 5: Generations from different models

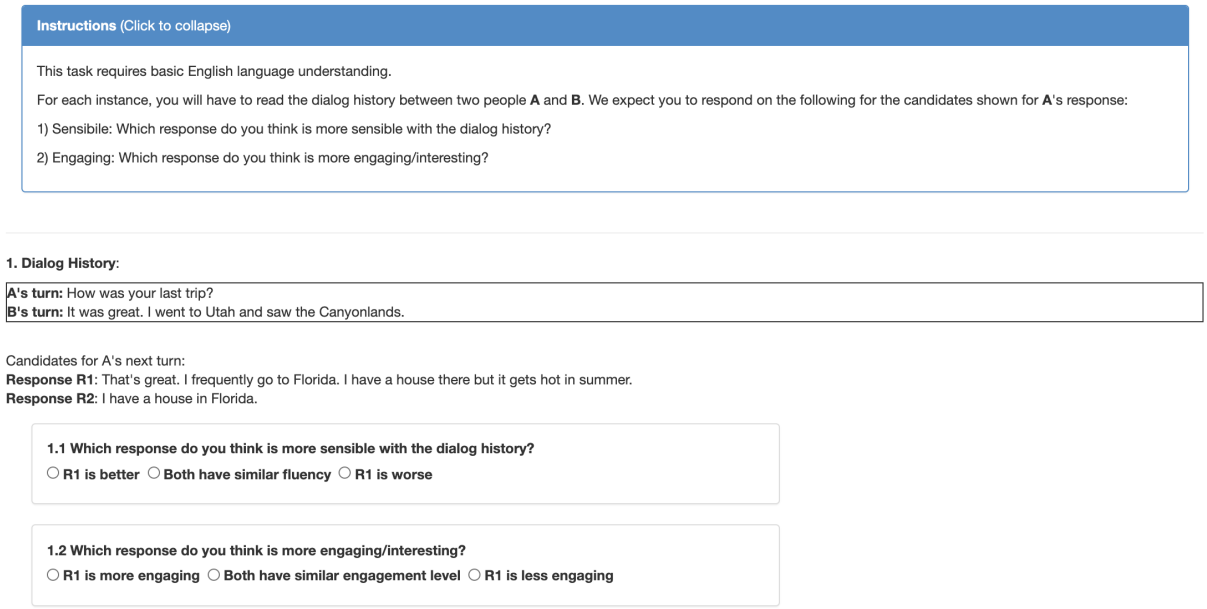


Figure 2: Human evaluation setup for pairwise comparison between PABST and another baseline

Beyond Laurel/Yanny: An Autoencoder-Enabled Search for Polyperceivable Audio

Kartik Chandra
Stanford University
kach@cs.stanford.edu

Chuma Kabaghe
Stanford University
chuma@alumni.stanford.edu

Gregory Valiant
Stanford University
gvaliant@cs.stanford.edu

Abstract

The famous "laurel/yanny" phenomenon references an audio clip that elicits dramatically different responses from different listeners. For the original clip, roughly half the population hears the word "laurel," while the other half hears "yanny." How common are such "polyperceivable" audio clips? In this paper we apply ML techniques to study the prevalence of polyperceivability in spoken language. We devise a metric that correlates with polyperceivability of audio clips, use it to efficiently find new "laurel/yanny"-type examples, and validate these results with human experiments. Our results suggest that polyperceivable examples are surprisingly prevalent, existing for $>2\%$ of English words.[1]

1 Introduction

How robust is human sensory perception, and to what extent do perceptions differ between individuals? In May 2018, an audio clip of a man speaking the word "laurel" received widespread attention because a significant proportion of listeners confidently reported hearing *not* the word "laurel," but rather the quite different sound "yanny" (Salam and Victor, 2018). At first glance, this suggests that the decision boundaries for speech perception vary considerably among individuals. The reality is more surprising: almost everyone has a decision boundary between the sounds "laurel" and "yanny," without a significant "dead zone" separating these classes. The audio clip in question lies close to this decision boundary, so that if the clip is slightly perturbed (e.g. by damping certain frequencies or slowing down the playback rate), individuals switch from confidently perceiving "laurel" to confidently perceiving "yanny," with the exact point of switching varying slightly from person to person.

[1]This research was conducted under Stanford IRB Protocol 46430.

How common is this phenomenon? Specifically, what fraction of spoken language is "polyperceivable" in the sense of evoking a multimodal response in a population of listeners? In this work, we provide initial results suggesting a significant density of spoken words that, like the original "laurel/yanny" clip, lie close to unexpected decision boundaries between seemingly unrelated pairs of words or sounds, such that individual listeners can switch between perceptual modes via a slight perturbation.

The clips we consider consist of audio signals synthesized by the Amazon Polly speech synthesis system *with a slightly perturbed playback rate* (i.e. a slight slowing-down of the clip). Though the resulting audio signals are not "natural" stimuli, in the sense that they are very different from the result of asking a human to speak slower (see Section 5), we find that they are easy to compute and reliably yield compelling polyperceivable instances. We encourage future work to investigate the power of more sophisticated perturbations, as well as to consider natural, ecologically-plausible perturbations.

To find our polyperceivable instances, we (1) devise a metric that correlates with polyperceivability, (2) use this metric to efficiently sample candidate audio clips, and (3) evaluate these candidates on human subjects via Amazon Mechanical Turk. We present several compelling new examples of the "laurel/yanny" effect, and we encourage readers to listen to the examples included in the supplementary materials (also available online at https://theory.stanford.edu/~valiant/polyperceivable/index.html). Finally, we estimate that polyperceivable clips can be made for $>2\%$ of English words.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 593–598
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Method

To investigate polyperceivability in everyday auditory input, we searched for audio clips of single spoken words that exhibit the desired effect. Our method consisted of two phases: (1) sample a large number of audio clips that are likely to be polyperceivable, and (2) collect human perception data on those clips using Amazon Mechanical Turk to identify perceptual modes and confirm polyperceivability.

2.1 Sampling clips

To sample clips that were likely candidates, we trained a simple autoencoder for audio clips of single words synthesized using the Amazon Polly speech synthesis system. Treating the autoencoder's low-dimensional latent space as a proxy for *perceptual* space, we searched for clips that travel through more of the space as the playback rate is slowed from $1.0\times$ to $0.6\times$. Intuitively, a longer path through encoder space should correspond to a more dramatic change in perception as the clip is slowed down (Section 3 presents some data supporting this).

Concretely, we computed a score S proportional to the length of the curve swept by the encoder E in latent space as the clip is slowed down, normalized by the straight-line distance traveled: that is, we define $S(c) = \frac{\int_{r=1.0\times}^{0.6\times} ||dE(c,r)/dr||dr}{||E(c,0.6\times)-E(c,1.0\times)||}$. Then, with probability proportional to $e^{0.2 \cdot S}$, we importance-sampled 200 clips from the set of audio clips of the top 10,000 English words, each spoken by all 16 voices offered by Amazon Polly (spanning American, British, Indian, Australian, and Welsh accents, and male and female voices). The distributions of S in the population and our sample is shown in Figure 2.

Autoencoder details Our autoencoder operates on one-second audio clips sampled at 22,050 Hz, which are converted to spectrograms with a window size of 256 and then flattened to vectors in $\mathbb{R}^{90,000}$. The encoder is a linear map to $\mathbb{R}^{512}$ with ReLU activations, and the decoder is a linear map back to $\mathbb{R}^{90,000}$ space with pointwise squaring. We used an Adam optimizer with lr=0.01, training on a corpus of 16,000 clips (randomly resampled to between 0.6x and 1.0x the original speed) for 70 epochs with a batch size of 16 ($\approx$ 8 hours on an AWS c5.4xlarge EC2 instance).

2.2 Mechanical Turk experiments

Each Mechanical Turk worker was randomly assigned 25 clips from our importance-sampled set of 200. Each clip was slowed to either 0.9x, 0.75x, or 0.6x the original rate. Workers responded with a perceived word and a confidence score for each clip. We collected responses from 574 workers, all of whom self-identified as US-based native English speakers. This yielded 14,370 responses ($\approx$ 72 responses per clip).

Next, we manually reviewed these responses and selected the most promising clips for a second round with only 11 of the 200 clips. Note that because these selections were made by manual review (i.e. listening to clips ourselves), there is a chance we passed over some polyperceivable clips — this means that our computations in Section 3 are only a conservative lower bound. For this round, we also included clips of the 5 words identified by Guan and Valiant (2019), 12 potentially-polyperceivable words we had found in earlier experiments, and "laurel" as controls. We collected an additional 3,950 responses among these 29 clips ($\approx$ 136 responses per clip) to validate that they were indeed polyperceivable.

Finally, we took the words associated with these 29 clips and produced a new set of clips using each of the 16 voices, for a total of 464 clips. We collected 4,125 responses for this last set ($\approx$ 3 responses for each word/voice/rate combination).

3 Results

Are the words we found polyperceivable? To identify cases where words had multiple perceptual "modes," we looked for clusters in the distribution of responses for each of the 29 candidate words. Concretely, we treated responses as "bags of phonemes" and then applied K-means. Though this rough heuristic discards information about the order of phonemes within a word, it works sufficiently well for clustering, especially since most of our words have very few syllables (more sophisticated models of phonetic similarity exist, but they would not change our results).

We found that the largest cluster typically contained the original word and rhymes, whereas other clusters represented significantly different perceptual modes. Some examples of clusters and their relative frequency are available in Table 1, and the relative cluster sizes as a function of playback rate are shown in Figure 1. As the rate is perturbed,

Perceived sound	Playback rate		
	0.90×	0.75×	0.60×
laurel/lauren/moral/floral	0.86	0.64	0.19
manly/alley/marry/merry/mary	0.0	0.03	0.35
thrilling	0.63	0.47	0.33
flowing/throwing	0.34	0.50	0.58
settle	0.65	0.25	0.33
civil	0.32	0.64	0.48
claimed/claim/climbed	0.58	0.34	0.11
framed/flam(m)ed/friend/ find	0.33	0.52	0.43
leg	0.50	0.31	0.10
lake	0.46	0.34	0.14
growing/rowing	0.50	0.47	0.26
brewing/booing/boeing	0.19	0.23	0.26
third	0.40	0.10	0.10
food/foot	0.18	0.29	0.13
idly/ideally	0.38	0.30	0.03
natalie	0.25	0.27	0.09
fiend	0.22	0.34	0.32
themed	0.11	0.17	0.24
bologna/baloney/bellany	0.26	0.00	0.00
(good)morning	0.03	0.28	0.77
thumb	0.66	0.74	0.79
fem(me)/firm	0.06	0.10	0.12
frank/flank	0.72	0.96	0.43
strength	0.08	0.00	0.15
round	0.53	0.38	0.65
world	0.03	0.00	0.14

Table 1: Some polyperceivable words (bold) and their alternate perceptual modes (below). Each row gives representative elements from the mode, and the proportion of workers whose response fell in that mode.

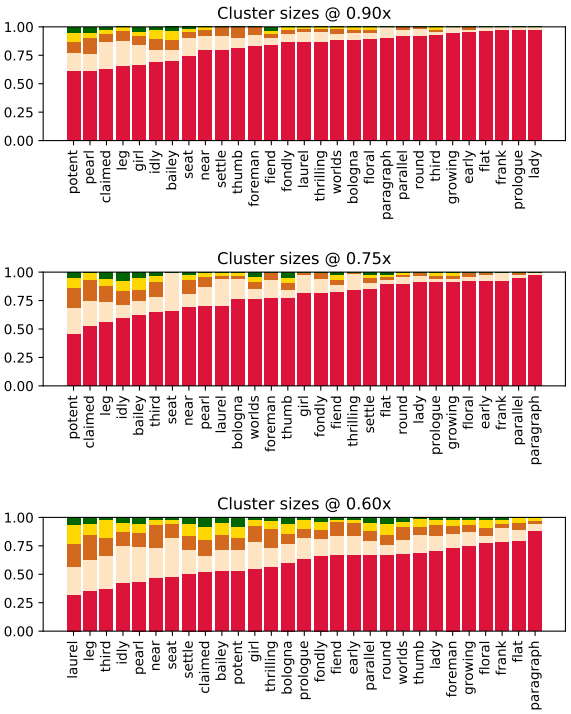


Figure 1: Relative cluster sizes across different playback rates. When the rate is slightly perturbed, the prevalence of alternate modes increases.

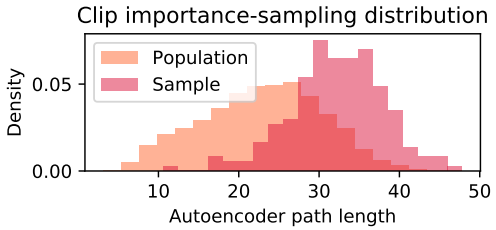


Figure 2: Distribution of path lengths (the S metric) in the population (top 10k English words, all 16 voices) and our sample of 200.

the prevalance of alternate modes among our clips increases.

How prevalent are polyperceivable words? Of our initial sample of 200 words, 11 ultimately yielded compelling demonstrations. To compute the prevalence of polyperceivable words in the population of the top 10k words, we have to account for the importance sampling weights we used when sampling in Section 2.1. After scaling each word's contribution by the inverse of the probability of including that word in our nonuniform sample of 200, we conclude that polyperceivable clips exist for at least 2% of the population: that is, of the 16 voices under consideration, at least one yields a polyperceivable clip for >2% of the top 10k English words.

We emphasize that this is a conservative lower bound, because it assumes that there were no other polyperceivable words in the 200 words we sampled, besides the 11 that we selected for the second round. We did not conduct an exhaustive search among those 200 words, instead focusing our Mechanical Turk resources on only the most promising candidates.

Is S a good metric? We consider the metric S to be successful because it allowed us to efficiently find several new polyperceivable instances. If the 200 words were sampled uniformly instead of being importance-sampled based on S, we would only have found 4 polyperceivable words in expectation (2% of 200). Thus, importance sampling increased our procedure's recall by almost 3×.

For a more quantitative understanding, we analyzed the relationship between "autoencoder path length" S and "perceptual path length" T. Our measure T of "perceptual path length" for a clip is *change in average distance between source word and response* as we slow the clip down from 0.75× to 0.6×. As with clustering above, distance

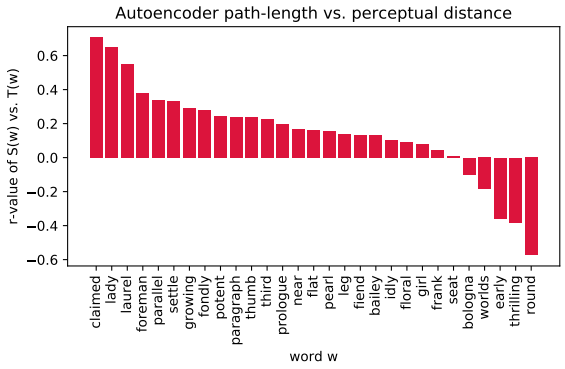


Figure 3: Correlation between S and T across the $n = 16$ voices for each of our 29 words. Nearly all words correlate positively, though with varying strengths (note that "laurel" correlates quite strongly).

is measured in bag-of-phonemes space. For each word, we computed the correlation between S and T among the 16 voices (both S and T vary significantly across voices). For all but 5 of our 29 words these metrics correlated positively, though with varying strength (Figure 3). This suggests that S indeed correlates with polyperceivability.

4 Discussion: Why study quirks of human perception in an ACL paper?

Perceptual instability in human sensory systems offers insight into ML systems. The question of what fraction of natural inputs lie close to decision boundaries for trained ML systems has received enormous attention. The surprising punchline that has emerged over the past decade is that *most* natural examples (including points in the training set) actually lie extremely close to unexpected decision boundaries. For most of these points, a tiny but carefully-crafted perturbation can lead the ML system to change the label. Such perturbations are analogous to the slight perturbation in playback speed for the polyperceivable clips we consider. In the ML literature, these perturbations, referred to as "adversarial examples" seem pervasive across complex ML systems (Szegedy et al., 2013; Goodfellow et al., 2014; Nguyen et al., 2015; Moosavi-Dezfooli et al., 2016; Madry et al., 2017; Raghunathan et al., 2018; Athalye et al., 2017).

While the initial work on adversarial examples focused on computer vision, more recent work shows the presence of such examples across other settings, including reinforcement learning (Huang et al., 2017), reading comprehension (Jia and Liang, 2017), and speech recognition (Carlini and Wagner, 2018; Qin et al., 2019). Studying perceptual illusions would provide a much-needed reference when evaluating ML systems in these domains. For vision tasks, for example, human vision provides the only evidence that current ML models are far from optimal in terms of robustness to adversarial examples. However, while humans are certainly not *as* susceptible to adversarial examples as ML systems, we lack quantified bounds on human robustness. More broadly, understanding which systems (both biological and ML) have decision boundaries that lie surprisingly close to many natural inputs may inform our sense of what settings are amenable to adversarially robust models, and what settings inherently lead to vulnerable classifiers.

Perceptual instability in ML systems offers insight into human sensory systems. Recent research on adversarial robustness of ML models has provided a trove of new tools and perspectives for probing classifiers and exploring the geometry of decision boundaries. These tools cannot directly be applied to study the decision boundaries of biological classifiers (e.g. we cannot reasonably do "gradient descent" on human subjects). However, using standard data-driven deep learning techniques to *model* human perceptual systems can allow us to apply these techniques by proxy.

An example can be found in the study of "transferability." Adversarial examples crafted to fool a specific model often also fool other models, even those trained on disjoint training sets (Papernot et al., 2016a; Tramèr et al., 2017; Liu et al., 2016). This prompts the question of whether adversarial examples crafted for an ML model might also transfer to *humans*. Recent surprising work by Elsayed et al. (2018) explores this question for vision. Humans were shown adversarial examples trained for an image classifier for $\approx$ 70ms, and asked to choose between the correct label and the classifier's (incorrect) predicted label. Humans selected the incorrect label more frequently when shown adversarial examples than when shown unperturbed images. Similarly, Hong et al. (2014) trained a low-dimensional representation of "perceptual space," and used the decision boundaries of the model to find images that confused human subjects.

5 Related work

An enormous body of work from cognitive sciences communities explores the quirks of human/animal sensory systems (Fahle et al., 2002). These works

often have the explicit goal of exploring isolated "illusions" that provide insights into our perceptual systems (Davis and Johnsrude, 2007; Fritz et al., 2005). However, there are few efforts to quantify the extent to which "typical" instances are polyperceivable or lie close to decision boundaries.

Miller (1981) studies the effect of speaking rate on how listeners perceive phonemes. The perceptual shifts studied therein are between phonetically adjacent perceptions (e.g. "pip" vs. "peep") rather than dramatically different perceptions (e.g. "laurel" vs. "yanny"). The "perturbation" of increasing human *speaking* rate is much more complex than simply linearly scaling the *playback* rate of an audio clip. Speaking-rate induced shifts also seem to hold more universally across voices, as opposed to the polyperceivable instances we examine.

6 Future work

Priming effects It is possible to use additional stimuli to alter perceptions of the "laurel/yanny" audio clip. For example, Bosker (2018) demonstrates the ability to control a listener's perception by "priming" them with a carefully crafted recording before the polyperceivable clip is played. Similarly, Guan and Valiant (2019) investigated the "McGurk effect" (McGurk and MacDonald, 1976), where what one "sees" affects what one "hears." The work estimated the fraction of spoken words that, when accompanied by a carefully designed video of a human speaker, would be perceived as significantly different words by listeners. Such phenomena raise questions about how our autoencoder-based method can be extended to search for "priming-sensitive" polyperceivability.

Security implications Just as adversarial examples for DNNs have security implications (Papernot et al., 2016b; Carlini and Wagner, 2017; Liu et al., 2016), so too might adversarial examples for sensory systems. For example, if a video clip of a politician happens to be polyperceivable, an adversary could lightly edit it with potentially significant ramifications. A thorough treatment of such security implications is left to future work.

7 Conclusion

In this paper, we leveraged ML techniques to study polyperceivability in humans. By modeling perceptual space as the latent space of an autoencoder, we were able to discover dozens of new polyperceivable instances, which were validated with Mechanical Turk experiments. Our results indicate that polyperceivability is surprisingly prevalent in spoken language. More broadly, we suggest that the study of perceptual illusions can offer insight into machine learning systems, and vice-versa.

Acknowledgements

We would like to thank Melody Guan for early discussions on this project, and the anonymous reviewers for their thoughtful suggestions. This research was supported by a seed grant from Stanford's HAI Institute, NSF award AF-1813049 and ONR Young Investigator Award N00014-18-1-2295.

References

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*.

Hans Rutger Bosker. 2018. Putting laurel and yanny in context. *The Journal of the Acoustical Society of America*, 144(EL503).

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE.

Matthew H Davis and Ingrid S Johnsrude. 2007. Hearing speech sounds: top-down influences on the interface between audition and speech perception. *Hearing research*, 229(1-2):132–147.

Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial examples that fool both computer vision and time-limited humans. In *Advances in Neural Information Processing Systems*, pages 3910–3920.

Manfred Fahle, Tomaso Poggio, Tomaso A Poggio, et al. 2002. *Perceptual learning*. MIT Press.

Jonathan B Fritz, Mounya Elhilali, and Shihab A Shamma. 2005. Differential dynamic plasticity of a1 receptive fields during multiple spectral tasks. *Journal of Neuroscience*, 25(33):7623–7635.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Melody Y. Guan and Gregory Valiant. 2019. A surprising density of illusionable natural speech. In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society (CogSci)*. cognitivesciencesociety.org.

Ha Hong, Ethan Solomon, Dan Yamins, and James J DiCarlo. 2014. Large-scale characterization of a universal and compact visual perceptual space. *space (P-space)*, 10(20):30.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Harry McGurk and John MacDonald. 1976. Hearing lips and seeing voices. *Nature*, 264(5588):746–748.

Joanne L Miller. 1981. Effects of speaking rate on segmental distinctions. *Perspectives on the study of speech*, pages 39–74.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016a. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016b. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE.

Yao Qin, Nicholas Carlini, Ian Goodfellow, Garrison Cottrell, and Colin Raffel. 2019. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International Conference on Machine Learning (ICML)*.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.

Maya Salam and Daniel Victor. 2018. Yanny or laurel? how a sound clip divided america. *The New York Times*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*.

Don't Let Discourse Confine Your Model: Sequence Perturbations for Improved Event Language Models

Mahnaz Koupaee[1], **Greg Durrett**[2], **Nathanael Chambers**[3], **Niranjan Balasubramanian**[1]

[1] Stony Brook University, [2] The University of Texas at Austin, [3] United States Naval Academy
[1]{mkoupaee,niranjan}@cs.stonybrook.edu
[2]gdurrett@cs.utexas.edu, [3]nchamber@usna.edu

Abstract

Event language models represent plausible sequences of events. Most existing approaches train autoregressive models on text, which successfully capture event co-occurrence but unfortunately constrain the model to follow the *discourse order* in which events are presented. Other domains may employ different discourse orders, and for many applications, we may care about different notions of ordering (e.g., temporal) or not care about ordering at all (e.g., when predicting related events in a schema). We propose a simple yet surprisingly effective strategy for improving event language models by perturbing event sequences so we can relax model dependence on text order. Despite generating completely synthetic event orderings, we show that this technique improves the performance of the event language models on both applications and out-of-domain events data.

1 Introduction

Event-level language models (LMs) provide a way to reason about events, and to approximate schematic and script-like knowledge (Schank and Abelson, 1977; Balasubramanian et al., 2013; Nguyen et al., 2015) about them (Modi and Titov, 2014; Pichotta and Mooney, 2016; Weber et al., 2018). These models aim to learn high-level representations of complex events (e.g., an arrest) and possibly their entity roles from raw text (e.g., a suspect). However, a major limitation is their reliance on the *discourse* order of event mentions when training the LM. Although powerful, these event LMs capture information we don't want in true world knowledge. For instance, a script of events may be weakly ordered in real life, but the system instead learns to strongly rely on the text order in which the events were described. Figure 1 shows an example where discourse and actual temporal order are different: a model trained on newswire may learn the pattern on the left from obituaries, but will fail to generalize to biographical or other narrative descriptions of someone's life.

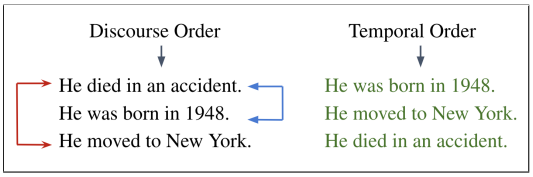


Figure 1: Example of an event schema for which the discourse order is different from the temporal order.

In this paper, we aim to improve event-level LMs in order to make them more suitable for general knowledge learning. While a range of possible modifications to the model can be imagined, such as set transformers (Lee et al., 2019), we want to leverage autoregressive pre-trained LMs. We instead find that we can encode the necessary invariances via data augmentation: namely, we apply a set of event sequence perturbations to sequences in the training data to relax the model's dependence on discourse order. By considering the next event based on shuffled sequences of events, we encourage the model to treat the input more as a set of events rather than strictly as a discourse sequence.

Surprisingly, despite our disruption of discourse order, experiments show how perturbations can improve event language modeling of text, particularly when evaluating the model on other domains which present events in different orders (e.g., novels or blogs present data in more of a "narrative" fashion than news datasets common in NLP (Yao and Huang, 2018)). Our experiments evaluate accuracy on the Inverse Narrative Cloze task on in-domain newswire, as well as out-domain novels and blogs[1].

[1]The code and data is available at https://github.com/StonyBrookNLP/elm-perturbations

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 599–604
August 1–6, 2021. ©2021 Association for Computational Linguistics

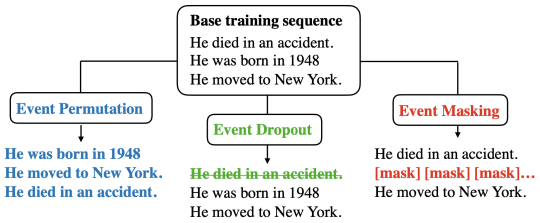


Figure 2: Sequence perturbations strategies.

2 Perturbing Discourse Sequences

Event language modeling tasks are typically defined over sequences of events as they appear in text. The events can be represented either as a sequence of words annotated with predicate-argument structure (e.g., semantic roles (Pichotta and Mooney, 2016), Open IE tuples (Weber et al., 2018; Rudinger et al., 2015) or with compositional embeddings (Modi, 2016). Generative models are trained to predict subsequent events in a sequence conditioning on previously observed events. Naturally, these models learn the order in which events appeared in text (Manshadi et al., 2008).

However, relying on discourse order may not be necessary and can potentially limit generalization of event LMs. For some event related tasks such as schema learning (Weber et al., 2018), the discourse order is not directly relevant. For other tasks such as event ordering (Pustejovsky et al., 2003; Chambers et al., 2014; Wang et al., 2018), temporal or logical order of events is most critical – discourse order, at best, is a noisy proxy. In fact, the first systems for schema learning were noticeably *not* language models (Mooney and DeJong, 1985; Chambers and Jurafsky, 2009, 2011). We introduce three simple perturbation techniques shown in Figure 2 that relax the reliance on discourse sequences.

2.1 Event Permutation

One way to reduce reliance on discourse order is to expose the model to random permutations of the input sequences, as shown in Figure 2. Using all possible permutations of a sequence is impractical, so we introduce three specific shuffles that force the model to pay attention to long-term dependencies and avoid the over-reliance on local dependencies/order:

- Reversed order: given a set of events as `ABCD`, the reverse of the sequence is created as `DCBA`.
- Concatenation of events in the odd positions followed by the even positions of the sequence: the permuted sequence is `BDAC`.
- Concatenation of event tuples in the odd positions followed by those in the even positions of the *reverse* order of the original sequence. The new sequence is: `CADB`

These shuffle patterns were selected to minimize the chance of repetition across permutations.

2.2 Event Dropout

We also consider event dropout as another perturbation to the original discourse sequence. For each sequence, we remove a small random subset of events (Event Dropout in Figure 2). We create multiple reduced sequences for each original sequence. The reduced sequences are treated in the same way as the original sequences for training the model. This perturbation is a type of regularization against overfitting on any specific event in a sequence, much like standard dropout procedures.

2.3 Event Masking

When dropping events, we can provide additional information to the model about where events were dropped. This forces the model to capture longer-term dependencies among events in the sequence. We randomly select a number of event tuples and replace their tokens with a `<mask>` token (Masking in Figure 2). For each sequence in the training set, we generate its masked sequences with each having a fixed proportion of its events masked.

3 Experimental Setup

Data We train event language models on the Annotated NYT corpus using Open IE event tuples extracted by Ollie (Schmitz et al., 2012). The dataset contains a total of around 1.8 million articles. After preprocessing steps, 1,467,366 articles are used as the training set, 6k articles as test set and 4k articles as the dev set. Each event is a 4-tuple `(v,s,o,p)` containing the verb, subject, object and preposition. We follow the same preprocessing steps outlined in Weber et al. (2018) to create event sequences.

The components of the events (the verb, subject, etc.) are all individual tokens, and are treated like normal text. For example, the events (truck packed with explosives), (police arrested suspect), would be given to the model as: packed truck explosives with [TUP] arrested police suspect _NULL_, where

NULL is the null preposition token and [TUP] is a special separator token between events.

Each document is first partitioned into segments of four sentences each. All events extracted from each segment are concatenated (in discourse order) to form an event sequence. This is a simple heuristic to avoid considering event sequences that can drift or connect otherwise unrelated events. Tuples with common verbs (is, are, be, ...) and repeating predicates are also ignored.

The training, development, and test splits have 7.1M, 19K, and 29K event sequences respectively. During training, depending on the perturbation strategy used, a number of sequences are added to the initial sets. The numbers are hyperparameters, selected differently for each model. Details are given in the following sections.

Autoregressive Models Our baseline autoregressive event LM is a pretrained GPT-2 model (Radford et al., 2019) fine-tuned on the event sequences.

Once the perturbations are applied to the original sequence, the modified sequence is used as both the input and the output of the model. We trained variants of GPT-2 with different sequence perturbations as shown in Figure 2 in our experiments. For the dropout and masked versions, we created $n/3$ new sequences with n being the number of events in the sequence. Each sequence has $n/3$ of its events either dropped or masked.

Autoencoding Models We use HierarchicAl Quantized Autoencoder (HAQAE) (Weber et al., 2018) as a strong autoencoding model. HAQAE is an LSTM-based autoencoder, which uses a hierarchical latent space to model event sequences. HAQAE uses categorical global latent variables to represent a tree-structured hierarchy which allow it to model different types of schemas and their possible tracks. Different levels of this hierarchical structure capture different levels of features of the schemas.

For training the HAQAE model, instead of reconstructing a perturbed sequence, we explore a denoising style training objective, where we *only* perturb the input part of the sequence keeping the output the same as the original. Our hypothesis is that these models learn a perturbation-invariant latent space representation in both cases, which will help break the dependence on discourse order. We use the denoising variant in our experiments as it worked better than the standard reconstruction objective in our initial experiments.

	Type of System	PPL Val	PPL Test	INC Val	INC Test
	Random Baseline	-	-	16.60	16.60
Auto regressive	RNNLM	91.84	90.92	25.30	26.30
	GPT-2 Baseline	85.13	84.13	26.80	28.30
	GPT-2 Masked	87.96	87.26	26.30	27.10
	GPT-2 Dropout	83.46	82.56	26.70	27.70
	GPT-2 Permuted	**83.18**	**82.26**	**27.45**	**28.90**
Auto encoding	HAQAE-Baseline	142.22	140.89	31.80	33.85
	HAQAE-Masked	148.07	147.03	33.80	36.80
	HAQAE-Dropout	**122.69**	**122.30**	31.25	32.25
	HAQAE-Permuted	143.39	142.07	**34.75**	**38.55**

Table 1: Perplexity and the accuracy of Inverse Narrative Cloze task. Lower is better for perplexity while higher is better for INC.

For each sequence in the permutation model, we generated permuted sequences for 10% of the original sequences. As for the dropout and masked models, we created $n/4$ new sequences with n being the number of events in the sequence. Each sequence has $n/3$ of its events either dropped or masked. Preliminary experiments showed little difference between using all the data vs a subset.

Models Hyperparameters The GPT-2 model uses the implementation from Huggingface library (Wolf et al., 2020) using a pre-trained gpt-2 small model and tokenizer. Adam optimizer (Kingma and Ba, 2014) is used with an initial learning rate of $6.25e-5$.

The HAQAE model uses 5 discrete latent variables. Each variable can initially take on K = 512 values, with an embeddings dimension of 256. The encoder is a bidirectional, single layer RNN with GRU cell (Cho et al., 2014) with a hidden dimension of size 512. The embeddings size is 300 which are initialized with pretrained GloVe (Pennington et al., 2014) vectors. The decoder is also a single layer RNN with GRU cells with a hidden dimension of 512 and 300 dimensional word embeddings (initialized) as inputs. All experiments use a vocabulary size of 50k. Adam optimizer with a learning rate of 0.0005 is used.

4 Evaluation

We ran different experiments to answer the following questions:

How do sequence perturbation techniques improve event language modeling? We evaluate perplexity as is standard in Table 1, but aside from

System	Blogs	Novels	News
HAQAE-baseline	24.31	25.10	32.25
HAQAE-permuted	**31.95**	**28.45**	**38.75**

Table 2: INC accuracy on external data

Legitimate Sequence	Confounding Sequence
↓	↓
he issued estimates	he issued estimates
he received extension	homes assess ability
candidate pledged before primary	department specify information
return release in august	provisions govern training
griffin president of investments	promise of laws unfulfilled
lauder pay income	promise remains unfulfilled

Figure 3: A legitimate sequence and its confounding.

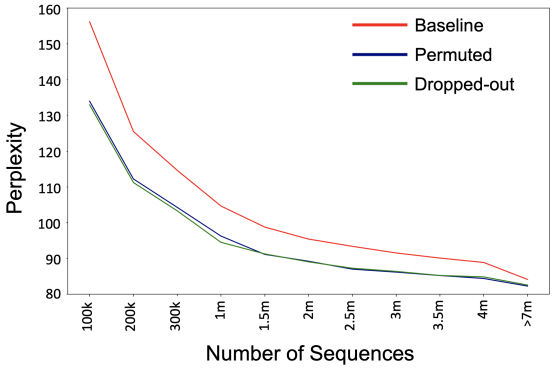


Figure 4: Perplexity of different GPT-2 models with respect to the number of training sequences.

perplexity, we want to see how well event LMs capture schematic knowledge. We thus evaluate on the inverse narrative cloze (INC) task (Weber et al., 2018). Given the first event from an original discourse sequence and a set of candidate event sequences, the task is to identify the true event sequence completion. This evaluation is closer to our ultimate goal: identifying realistic event schemas rather than discourse-focused metrics like perplexity.

The INC evaluation starts with a gold sequence of events from a real document, and then includes 5 other event sequences pulled from confounding documents. You insert the first gold event artificially at the start of each of these. The gold event sequence should have high probability compared to the confounding event sequences. Figure 3 shows a gold sequence and one confounding sequence generated for it. The six sequences are ranked based on the probabilities assigned by the model, and then the accuracy is the number of predictions where the gold sequence is ranked first. A random model will uniformly choose one among the six sequences and thus will have an accuracy of 16.6%.

The perplexity[2] and the INC accuracy of different variants of both autoregressive and autoencoding models are shown in Table 1.

Using sequence perturbations improves the INC accuracy on both test and validation sets for both categories of models. Further, the sequence perturbations gain in terms of INC accuracy is much higher with HAQAE.

[2]For autoencoders we report generative perplexity with the KL-term, while the original paper (Weber et al., 2018) has the lower reconstructive perplexity without the KL-term.

How do models trained with perturbation techniques perform on out-of-domain data? The NYT corpus used for training the models in this study is newswire. The journalistic writing style does not always follow the temporal ordering of events, but represents the events in various orders going backwards or forward in time. One might argue that the reason the sequence perturbations work better in terms of INC accuracy is that the events extracted from news do not necessarily follow the temporal order and therefore the perturbations will not create an issue. To show the effectiveness of our approach, we evaluated the performance of our models on the event sequences extracted from narratives coming from different domains: novels, blogs and news (Yao and Huang, 2018).

We used the OpenIE extraction system in a similar fashion to extract the event tuples from the narrative sequences. We used our best-performing model from the previous section and with no fine-tuning applied the models to see how our sequence perturbations performed in terms of INC accuracy on these narrative texts. The results of this analysis are presented in Table 2. The numbers show that the proposed sequence perturbations perform better on out-of-domain data (with explicit temporal links) compared to the baseline model.

How effective are the sequence perturbation techniques with respect to the number of training instances? Our sequence perturbations can be seen as data augmentation strategies which will help models learn new aspects of data that can not be learned from the original sequences. As the number of training samples increases, the model has more opportunities to learn these aspects. Therefore, the sequence perturbations will be more useful for domains with fewer training samples.

	seed	generated events	ppl(g—s)	ppl(g—ps)
HAQAE-Baseline	people reported fire, fire spread to forest	people died in fire, fire caused fires	4.49	6.49
	fire spread to forest, people reported fire	person spokesman for department, firefighters taken to hospital	5.21	6.30
HAQAE-permuted	people reported fire, fire spread to forest	fires began today, people working in area	5.75	5.58
	fire spread to forest, people reported fire	fires began today, people working in area	5.58	5.75

Table 3: Generated schemas for two-event seeds. The second row for each model shows the generated schemas for permuted seed events. ppl(g—s) and ppl(g—ps) are the perplexity of generated events given the seed events and the perplexity of events given permuted seeds. The lower the difference the more robust the model is to permutations.

	seed	generated events
HAQAE-baseline	fire spread to neighborhood, people reported fire	people fire to floor, person spokesman for department
	fire spread to forest, people reported fire	firefighters fire to floor, person spokesman for department
HAQAE-permuted	fire spread to neighborhood, people reported fire	Fire spread through floors, fire came from floor
	fire spread to forest, people reported fire	fires began today, people working in area

Table 4: Generated schemas for two-event seeds. The second event is the same while the first event shows a different branch.

We plotted the perplexity with respect to the number of training sequences for the GPT-2 baseline system as well as permuted and dropout models. As can be seen in Figure 4, the gap between the perplexity scores are higher when the number of sequences are lower. This observation suggests that our approach will result in better language models for domains with limited data.

How do schemas generated by different models differ from each other? We generated schemas for 46 two-event seeds using the HAQAE baseline and permuted models. We wanted to see how the generated schemas differ in two different aspects: First, for each seed, we permuted the events and generated schemas for both models. We expect the permuted model to have less variation in generating events for original and permuted seeds. We calculated the perplexity of the generated events for both the original order of events as well as the permuted order. Table 3 shows an example of such scenario where the HAQAE-permuted model has lower variation in perplexity for permuted seed events.

Second, we want to see how dependent the generation is upon the most recent event in the sequence. We generated schemas for two-event seeds in which the last event is the same while the first event indicates a different path. Table 4 shows an example where the permuted model generates more diverse events.

5 Conclusion

We proposed a set of simple sequence perturbations to relax the model's reliance on the discourse order of event mentions for event language modeling. By predicting the next event based on perturbed sequences, the model is encouraged to treat the input as a *set* of events. Our experiments show that these perturbations can improve identifying event schemas measured by INC accuracy both on in-domain and out-of-domain data.

Acknowledgments

We would like to thank Noah Weber for providing helpful directions on the HAQAE model. This material is based on research that is supported by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes.

References

Niranjan Balasubramanian, Stephen Soderland, Oren Etzioni, et al. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 976–986.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR.

Mehdi Manshadi, Reid Swanson, and Andrew S Gordon. 2008. Learning a probabilistic model of event sequences from internet weblog stories. In *FLAIRS Conference*, pages 159–164.

Ashutosh Modi. 2016. Event embeddings for semantic script modeling. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 75–83.

Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 49–57.

Raymond J Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *IJCAI*, pages 681–687.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Karl Pichotta and Raymond J Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Rob Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003. The TimeBank corpus. *Proceedings of Corpus Linguistics*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals, and understanding: an inquiry into human knowledge structures.

Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534.

Su Wang, Eric Holgate, Greg Durrett, and Katrin Erk. 2018. Picking apart story salads. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1465, Brussels, Belgium. Association for Computational Linguistics.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical quantized representations for script generation. *arXiv preprint arXiv:1808.09542*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenlin Yao and Ruihong Huang. 2018. Temporal event knowledge acquisition via identifying narratives. *arXiv preprint arXiv:1805.10956*.

The Curse of Dense Low-Dimensional Information Retrieval for Large Index Sizes

Nils Reimers and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technical University of Darmstadt
www.ukp.tu-darmstadt.de

Abstract

Information Retrieval using dense low-dimensional representations recently became popular and showed out-performance to traditional sparse-representations like BM25. However, no previous work investigated how dense representations perform with large index sizes. We show theoretically and empirically that the performance for dense representations decreases quicker than sparse representations for increasing index sizes. In extreme cases, this can even lead to a tipping point where at a certain index size sparse representations outperform dense representations. We show that this behavior is tightly connected to the number of dimensions of the representations: The lower the dimension, the higher the chance for false positives, i.e. returning irrelevant documents.

1 Introduction

Information retrieval traditionally used sparse representations like TF-IDF or BM25 to retrieve relevant documents for a given query. However, these approaches suffer from the lexical gap problem (Berger et al., 2000).

To overcome this issue, dense representations have been proposed (Gillick et al., 2018): Queries and documents are mapped to a dense vector space and relevant documents are retrieved e.g. by using cosine-similarity. Out-performance over sparse lexical approaches has been shown for various datasets (Gillick et al., 2018; Guo et al., 2020; Guu et al., 2020; Gao et al., 2020).

Previous work showed the out-performance for fixed, rather small indexes. The largest dataset where it has been shown is the MS Marco (Bajaj et al., 2018) passage retrieval dataset, where retrieval is done over an index of 8.8 million text passages. However, in production scenarios, index sizes quickly reach 100 millions of documents.

We show in this paper, that the performance for dense representations can decrease quicker for increasing index sizes than for sparse representations. For a small index of e.g. 100k documents, a dense approach might clearly outperform sparse approaches. However, with a larger index of several million documents, the sparse approach can outperform the dense approach.

We show theoretically and empirically that this effect is closely linked to the number of dimensions for the representations: Using fewer dimensions increases the chances for false positives. This effect becomes more severe with increasing index sizes.

2 Related Work

A common choice for dense retrieval is to fine-tune a transformer network like BERT (Devlin et al., 2018) on a given training corpus with queries and relevant documents (Guo et al., 2020; Guu et al., 2020; Gao et al., 2020; Karpukhin et al., 2020; Luan et al., 2020). Recent work showed that combining dense approaches with sparse, lexical approaches can further boost the performance (Luan et al., 2020; Gao et al., 2020). While the approaches have been tested on various information and question answering retrieval datasets, the performance was only evaluated on fixed, rather small indexes. Guo et al. (2020) evaluated approaches for eight different datasets having index sizes between 3k and 454k documents.

We are not aware of previous work that compares sparse and dense approaches for increasing index sizes and the connection to the dimensionality. The only work we are aware of that systematically studies the encoding size for dense approaches is (Luan et al., 2020), but they only studied the connection to the document length.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 605–611
August 1–6, 2021. ©2021 Association for Computational Linguistics

3 Theory

Dense retrieval approaches map queries and documents[1] to a fixed size dense vector. The most relevant documents for a given query can then be found using cosine-similarity.

Using as few dimensions as possible is desirable, as it decreases the memory requirement to store (an index) of millions of vectors and leads to faster retrieval. However, as we show, lower-dimensional representations can have issues with large indices.

Given a query vector $q \in \mathbb{R}^k$, we search our index of document vectors $d_1, ..., d_n \in \mathbb{R}^k$ for the documents that maximizes:

$$\text{cossim}(q, d_i) = \cos(\theta) = \frac{q \cdot d_i}{\|q\| \, \|d_i\|}$$

Note: In the following we just show the case for cosine similarity. The proof extends to other similarity functions like dot-product and any p-norm (Manhatten, Euclidean) as long as the vector space is finite. A finite n-dimensional vector space can be mapped to an $n+1$-dimensional vectors space with vectors of unit length. In that case, dot-product in n dimensions is equivalent to cosine-similarity in $n+1$ dimensions. Similar, any p-norm in n dimensions can be re-written as cosine-similarity in $n+1$ dimensions.

Theorem: The probability for false positives (I) increases with the index size n and (II) with the decreasing dimensionality k.

Proof (I): Given a query q and the relevant document d_r. For simplicity, we assume only a single relevant document. If multiple documents are relevant, we consider only the one with the highest cosine similarity. In order that no false positive is returned, $\text{cossim}(q, d_r)$ must be greater than $\text{cossim}(q, d_i)$ for all $i \neq r$. Assume the possible vectors are independent. Then, the probability for a false positive is

$$P(\text{false positive}) = 1-(1-P(\text{false positive}_i))^{n-1}$$

for an index with $n-1$ negative elements and $P(\text{false positive}_i)$ the probability that a single element is a false positive, i.e. $\text{cossim}(q, d_i) > \text{cossim}(q, d_r)$.

Proof (II): While the previous proof is straightforward, that the chance of false positives increases with larger index sizes, the more interesting aspect is the relation to the dimensionality, i.e., what is the probability $P(\text{false positive}_i)$ = $P(\text{cossim}(q, d_i) > \text{cossim}(q, d_r))$ for a random d_i? We show that this probability decreases with more dimensions.

Without loss of generality, we assume that the vectors are of unit length. The vectors are then on an k-dimensional sphere with radius 1. A false positive happens if $\text{cossim}(q, d_i) > \text{cossim}(q, d_r)$, or, equivalent if $1-\text{cossim}(q, d_i) < 1-\text{cossim}(q, d_r)$. I.e., we intersect the sphere in k dimensions with a hyperplane in $k-1$ dimensions. The area of the cut-off portion is defined by $1-\text{cossim}(q, d_r)$. All vectors within the cut-off portion (i.e. spherical cap) are false positives. The probability that a random vector will be returned as false positive is:

$$P(\text{false positive}_i) = A_{cap}/A_{sphere}$$

with A_{cap} the surface area of the spherical cap and A_{sphere} the surface area of the sphere in k dimensions. Define the surface area of the sphere in k dimensions as A_k, then the surface area of A_{cap} is (Li, 2011):

$$A_{cap} = \frac{1}{2} A_k I_{sin^2\theta}\left(\frac{k-1}{2}, \frac{1}{2}\right)$$

with $I_x(a, b)$ the regularized incomplete beta function and θ the polar angle, i.e. the angle between q and the relevant document d_r. Hence:

$$P(\text{false positive}_i) = \frac{1}{2} I_{sin^2\theta}\left(\frac{k-1}{2}, \frac{1}{2}\right) \quad (1)$$

For constant cosine similarity between query q and relevant document d_r, $I_{sin^2\theta}\left(\frac{k-1}{2}, \frac{1}{2}\right)$ is a monotonically decreasing function with increasing dimension k. In conclusion, more dimensions decrease the probability for false positives.

Combining (I) and (II) shows that a low dimensional representation might work well for small index sizes. However, with more indexed documents, the probability of false positives increases faster for low dimensional representations than for higher dimensional representations. Hence, at some index size, higher dimensional representations might outperform the lower-dimensional representation.

4 Empirical Investigation

In the proof, we have assumed that vectors are independent and uniformly distributed over the space, which gives us a lower bound on the false positive rate. However, in practice, dense representations are neither independent nor uniformly

[1] We use *document* as a cover-term for text of any length.

distributed. As shown in (Ethayarajh, 2019; Li et al., 2020), dense representations derived from pre-trained Transformers like BERT map to an anisotropic space, i.e., the vectors occupy only a narrow cone in the vector space. This drastically increases the chance that an irrelevant document is closer to the query embedding than the relevant document. Hence, we study how actual dense models are impacted by increasing index sizes and lower-dimensional representations.

4.1 Dataset

We conduct our experiments on the MS MARCO passage dataset (Bajaj et al., 2018). It consists of over 1 million unique real queries from the Bing search engine, together with 8.8 million paragraphs from heterogeneous web sources. Most of the queries have only 1 passage judged as relevant, even though more can exist. The development set consists of 6980 queries and the performance is evaluated using mean reciprocal rank MRR@10.

To better compare the relative performance differences, we compute a rank-aware error rate:

$$\text{Err} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{1}{\text{rank}_i}\right)$$

with rank_i being the rank of the relevant document for the i-th query. To be compatible with MRR@10, we set $\text{rank}_i = \infty$ for $\text{rank}_i > 10$. We then define the relative error rate as $\text{Err}_{\text{Dense}}/\text{Err}_{\text{BM25}}$. A relative error rate of 50% indicates that the dense approach makes only 50% of the errors compared to BM25 retrieval.

4.2 Model

For sparse, lexical retrieval, we use ElasticSearch, which is based on BM25. For dense retrieval, we use a DistilRoBERTa-base model (Sanh et al., 2020) as a bi-encoder: The query and the passage are passed independently to the transformer model and the output is averaged to create fixed-sized representations. We train this using InfoNCE loss (van den Oord et al., 2018):

$$L = -\log \frac{\exp(\tau \cdot \text{cossim}(q, p_+))}{\sum_i \exp(\tau \cdot \text{cossim}(q, p_i))}$$

with q the query, p_+ the relevant passage. We use in-batch negative sampling and use the other passages in a batch as negative examples. We found that $\tau = 20$ performs well. We train the model in two setups: 1) only with random (in-batch) negatives, and 2) we provide for each query additionally one hard-negative passage. We use the hard-negative passages provided by the MS MARCO dataset, which were retrieved using lexical search. Models are trained with a batch size of 128 with Adam optimizer and a learning rate of $2e-5$.

DistilRoBERTa produces representations with 768 dimensions. We also experiment with lower-dimensional representations. There, we added a linear projection layer on-top of the mean pooling operation to down-project the representation to either 128 or 256 dimensions. Dense retrieval is performed using cosine similarity with exact search.

Models were trained using the SBERT framework (Reimers and Gurevych, 2019).[2]

5 Experiments

First, we study the impact of increasing index sizes with real text passages. Then, we study the performance when random noise is added.

5.1 Increasing Index Size

In the first experiment, we start with an index that only contains the 7433 relevant passages for the 6980 queries. Then, we add step-wise randomly selected passages from the MS MARCO corpus to the index until all 8.8 million passages are indexed.

Model	10k	100k	1M	8.8M
BM25	79.93	63.88	40.14	17.56
Trained without hard negatives				
128 dim	87.50	68.63	39.76	15.71
256 dim	88.82	70.79	41.74	17.08
768 dim	88.99	71.06	42.24	17.34
Trained with hard negatives				
128 dim	90.32	77.92	54.45	27.34
256 dim	91.10	78.90	55.51	28.16
768 dim	91.48	79.42	56.05	28.55

Table 1: Dev performance (MRR@10 $\times 100$) on MS MARCO passage dataset with different index sizes. Higher score = better.

Table 1 shows the MRR@10 performance for the different systems. Increasing the index naturally decreases the performance for all systems, as retrieving the correct passages from a larger index is more challenging. The dense approach trained without hard negatives clearly outperforms BM25 for an index with 10k - 1M entries, but with all 8.8 million passages it performs worse than BM25.

Table 2 shows the relative error rate in comparison to BM25 retrieval. For small index sizes, we

[2] https://www.SBERT.net

Model	10k	100k	1M	8.8M
Trained without hard negatives				
128 dim	62.3	86.8	100.6	102.2
256 dim	55.7	80.9	97.3	100.6
768 dim	54.9	80.1	96.5	100.3
Trained with hard negatives				
128 dim	48.2	61.1	76.1	88.1
256 dim	44.3	58.4	74.3	87.1
768 dim	42.5	57.0	73.4	86.7

Table 2: Relative error rate (%) of dense approaches in comparison to BM25 retrieval. Lower score = better.

observe that dense approaches drastically reduce the error rate compared to BM25 retrieval. With increasing index sizes, the gap closes.

5.2 Index with Random Noise

MS MARCO is sparsely labeled, i.e., there is usually only a single passage labeled as relevant even though multiple passages would be considered as relevant by humans (Craswell et al., 2020). To avoid that the drop in performance is due to the retrieval of relevant, but unlabeled passages, we perform an experiment where we add random irrelevant noise to the index. Our index consists only of the relevant passages and a large fraction of irrelevant, randomly generated strings.[3]

We also evaluate the popular DPR system by Karpukhin et al. (2020), which is a BERT-based dense retriever trained on the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). We chose the NQ dev set, consisting of 1772 questions from Google search logs. DPR encodes the passage as `Title [SEP] Paragraph`. We create a random string for the paragraph and combine it with 1) a randomly generated string as title, 2) selecting randomly one of the over 6 Million real Wikipedia article titles, 3) selecting randomly one of the 1772 article titles found in the NQ dev set.

We count for how many queries a random string is ranked higher than the relevant passage. The results are shown in Table 3. We observe that BM25 does not rank any randomly generated passage higher than the relevant passage for the MS MARCO dataset. The chance that a random passage contains words matching the query is small.

For the dense retrieval models, we observe for quite a large number of queries that a random string passage is ranked higher than the relevant passage. As proven in Section 3, the error increases with larger index sizes and fewer dimensions.

[3]Strings are generated randomly using lowercase characters a-z and space.

Model	100k	1M	10M	100M
BM25	0.00%	0.00%	0.00%	0.00%
Dense without hard negatives - MS MARCO				
128 dim	2.71%	4.41%	6.69%	9.73%
256 dim	2.39%	4.03%	6.16%	9.04%
768 dim	2.13%	3.72%	5.77%	8.52%
Dense with hard negatives - MS MARCO				
128 dim	2.87%	4.20%	6.00%	8.11%
256 dim	2.45%	3.72%	5.59%	7.38%
768 dim	2.12%	3.32%	5.09%	7.03%
DPR (Karpukhin et al., 2020) - Natural Questions				
rnd title	0.17%	0.28%	0.34%	0.51%
all titles	2.48%	5.59%	9.31%	12.08%
dev titles	4.18%	5.36%	6.66%	8.01%

Table 3: Percentage of queries for which a random string passage is ranked higher than the relevant passage. 100k/1M/10M/100M indicates the number of random passages in the index.

For DPR, we observe an extreme dependency on the title. Having 100 million entries in the index with a real Wikipedia article title and a random paragraph, results in the retrieval of those for about 12.08% of all questions at the top position.

The error numbers far exceed the estimation from equation (1), confirming that the representations are not uniformly distributed over the complete vector space and are concentrated in a small space. In the appendix (Figure 1), we plot the representations for the queries, the relevant passages, and the random strings.

6 Conclusion

We have proven and shown empirically that the probability for false positives in dense information retrieval depends on the index size and on the dimensionality of the used representations. These approaches can even retrieve completely irrelevant, randomly generated passages with high probability. It is important to understand the limitations of dense retrieval:

1) Dense approaches work better for smaller, clean indexes. With increasing index size the difference to sparse approaches can decreases.

2) Evaluation results with smaller indexes cannot be transferred to larger index sizes. A system that is state-of-the-art for an index of 1 million documents might perform badly on larger indices.

3) The false positive rate increases with fewer dimensions.

4) The empirically found error rates far exceeded the mathematical lower-bound error rates, indicating that only a small fraction of the available vector space is effectively used.

Acknowledgments

This work has been supported by the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1 and grant GU 798/17-1) and has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268 v3*.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 192–199, New York, NY, USA. Association for Computing Machinery.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820 v2*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2020. Complementing Lexical Retrieval with Semantic Residual Embedding. *arXiv preprint arXiv:2004.13969*.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-End Retrieval in Continuous Space. *arXiv preprint arXiv:1811.08008*.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2020. MultiReQA: A Cross-Domain Evaluation for Retrieval Question Answering Models. *arXiv preprint arXiv:2005.02507*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv preprint arXiv:2002.08909*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

S. Li. 2011. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4:66–70.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, Dense, and Attentional Representations for Text Retrieval. *arXiv preprint arXiv:2005.00181*.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108 v4*.

A Plot of Random Noise Index

Figure 1 shows a two-dimensional plot of the 6980 development queries in the MS MARCO passage dataset, together with the 7433 passages that are marked as relevant and 7433 representations for randomly generated strings (using lowercase characters and space with a random length between 20 and 150 characters). The representation for the random strings are concentrated, but we still observe a significant overlap with the region for queries and relevant documents. This explains why random strings are retrieved for certain queries (Table 3). We use the dense model that was trained with hard negatives with 768 dimensions. UMAP (McInnes et al., 2018) is used for dimensionality reduction to 2 dimensions.

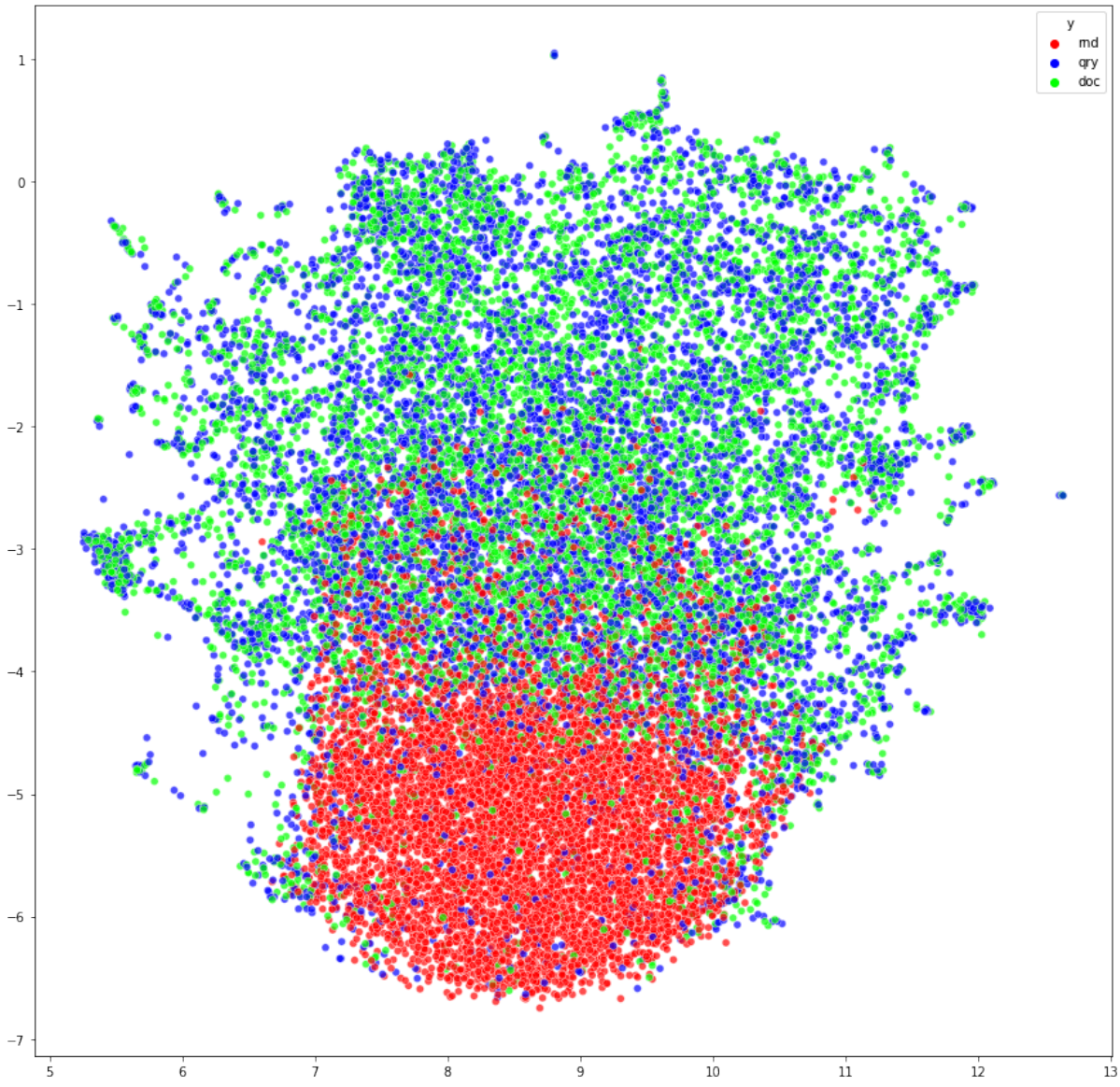


Figure 1: Plot of queries (blue), the relevant document (green) and representations from randomly generated strings (red). Dimensionality reduction via UMAP (McInnes et al., 2018). Model with hard negatives, 768 dimensions.

Cross-lingual Text Classification with Heterogeneous Graph Neural Network

Ziyun Wang[1]*, **Xuan Liu**[2*†], **Peiji Yang**[1] , **Shixing Liu**[1] , **Zhisheng Wang**[1]
Tencent, Shenzhen, China
[1]{billzywang, peijiyang, shixingliu, plorywang}@tencent.com
[2]lxstephenlaw@gmail.com

Abstract

Cross-lingual text classification aims at training a classifier on the source language and transferring the knowledge to target languages, which is very useful for low-resource languages. Recent multilingual pretrained language models (mPLM) achieve impressive results in cross-lingual classification tasks, but rarely consider factors beyond semantic similarity, causing performance degradation between some language pairs. In this paper we propose a simple yet effective method to incorporate heterogeneous information within and across languages for cross-lingual text classification using graph convolutional networks (GCN). In particular, we construct a heterogeneous graph by treating documents and words as nodes, and linking nodes with different relations, which include part-of-speech roles, semantic similarity, and document translations. Extensive experiments show that our graph-based method significantly outperforms state-of-the-art models on all tasks, and also achieves consistent performance gain over baselines in low-resource settings where external tools like translators are unavailable.

1 Introduction

The success of recent deep learning based models on text classification relies on the availability of massive labeled data (Conneau et al., 2017; Tian et al., 2020; Guo et al., 2020). However, labeled data are usually unavailable for many languages, and hence researchers have developed the setting where a classifier is only trained using a resource-rich language and applied to target languages without annotated data (Xu et al., 2016; Chen and Qian, 2019; Fei and Li, 2020). The biggest challenge is to bridge the semantic and syntactic gap between languages. Most existing methods explore the semantic similarity among languages, and learn a language-agnostic representation for documents from different languages (Chen et al., 2018; Zhang et al., 2020a). This includes recent state-of-the-art multilingual pretrained language models (mPLM) (Devlin et al., 2019; Conneau and Lample, 2019), which pretrain transformer-based neural networks on large-scale multilingual corpora. The mPLM methods show superior cross-lingual transfer ability in many tasks (Wu and Dredze, 2019). However, they do not explicitly consider syntactic discrepancy between languages, which may lead to degraded generalization performance on target languages (Ahmad et al., 2019; Hu et al., 2020).

On the other hand, there usually exists sufficient unlabeled target-language documents that come naturally with rich information about the language and the task. However, only a handful of previous researches have taken advantage of the unlabeled data (Wan, 2009; Dong and de Melo, 2019).

To integrate both semantic and syntactic information within and across languages, we propose a graph-based framework named Cross-Lingual Heterogeneous GCN (CLHG). Following the work of TextGCN (Yao et al., 2019), we represent all the documents and words as graph nodes, and add different types of information into the graph. We utilize mPLM to calculate the representation of all the nodes, and connect documents nodes with semantically similar ones to extract the knowledge in mPLM. Words are connected with documents based on the co-occurrences as in previous works. However, we choose to separate different word-doc edges by part-of-speech (POS) tags of words to inject some shallow syntactic information into the graph, as POS taggers are one of the most widely accessible NLP tools, especially for low-resource languages. In-domain unlabeled documents are added to the graph if available. To further absorb

*The first two authors contribute equally to this work.
†This work is done during Xuan Liu's internship at Tencent.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 612–620
August 1–6, 2021. ©2021 Association for Computational Linguistics

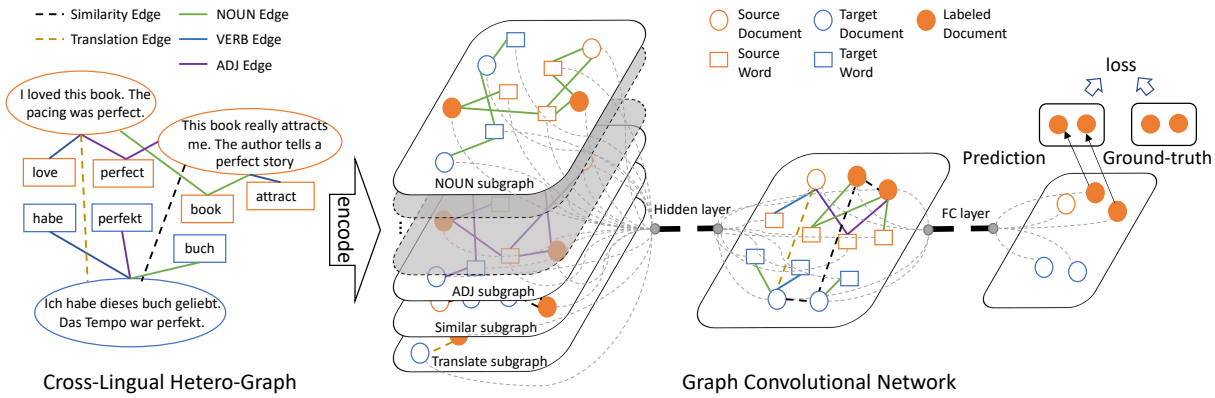


Figure 1: Illustration of our Cross-Lingual Heterogeneous GCN (CLHG) framework. For simplicity, only some POS tags are plotted in this graph. We recommend to view this figure in color as we use different colors to indicate different languages and edge types.

in-domain language alignment knowledge, we utilize machine translation to create translated text nodes. The text classification task is then formalized as node classification in the graph and solved with a heterogeneous version of Graph Convolutional Networks (Kipf and Welling, 2017). Our contributions are summarized as follows:

(1) We propose a graph-based framework to easily comprise heterogeneous information for cross-lingual text classification, and design multiple types of edges to integrate all these information. To the best of our knowledge, this is the first study to use heterogeneous graph neural networks for cross-lingual classification tasks. (2) We conduct extensive experiments on 15 tasks from 3 different datasets involving 6 language pairs. Results show that our model consistently outperforms state-of-the-art methods on all tasks without any external tool, and achieves further improvements with the help of part-of-speech tags and translations.

2 Related Works

Traditional methods for cross-lingual classification usually translate the texts (Wan, 2009) or the classifier model (Xu et al., 2016) with external aligned resources such as bilingual dictionaries (Andrade et al., 2015; Shi et al., 2010) or parallel corpora (Duh et al., 2011; Zhou et al., 2016; Xu and Yang, 2017). Recent works focus on learning a shared representation for documents of different languages, including bilingual word embeddings (Zou et al., 2013; Ziser and Reichart, 2018; Chen et al., 2018), common subword representations (Zhang et al., 2020a), and multilingual pretrained language models (mPLM) (Devlin et al., 2019; Conneau and Lample, 2019; Clark et al., 2020).

In the past few years, graph neural networks (GNN) have attracted wide attention, and become increasingly popular in text classification (Yao et al., 2019; Hu et al., 2019; Ding et al., 2020; Zhang et al., 2020b). These existing work mainly focus on monolingual text classification, except a recent work (Li et al., 2020) using meta-learning and graph neural network for cross-lingual sentiment classification, which nevertheless only uses GNN as a tool for meta-learning.

3 Method

In this section, we will introduce our CLHG framework, including how to construct the graph and how to solve cross-lingual text classification using heterogeneous GCN. In general, we first construct a cross-lingual heterogeneous graph based on the corpus and selected features, and next we encode all the texts with multilingual pre-trained language models, then we pass the encoded nodes to the heterogeneous GCN, each layer of which performs graph convolution on different subgraphs separated by different edge types, and aggregates the information together. Finally, the graph neural network outputs the predictions of doc nodes, which will be compared with groundtruth labels during training. Figure 1 shows the overall structure of the framework.

3.1 Graph Construction

Inspired by some previous works on GNN-based text classification (Yao et al., 2019; Hu et al., 2019), we construct the graph by representing both documents and words from the corpus in both languages

as graph nodes, and augment the corpus by including unlabeled in-domain documents from the target language. To extract more information of language alignments, we further use a publicly available machine translation API[1] to translate the documents in both directions. Then two categories of edges are defined in the graph.

Doc-word Edges. Like TextGCN (Yao et al., 2019), documents and words are connected by their co-occurrences. To inject syntactic information more than just co-occurrences, we add part-of-speech (POS) tags to the edges, since different POS roles have different importance in the classification tasks. Adjectives and adverbs are mostly decisive in sentiment classification, while nouns may play a more significant role in news classification. Therefore, we use POS taggers to tag each sentence and create different types of edges based on the POS roles of the words in the document, which could help GNN to learn different propagation patterns for each POS role.

Doc-doc Edges. To add more direct connections between documents, we include two types of document level edges. Firstly, we link each document with similar ones by finding K documents with the largest cosine similarity. The embeddings of the documents are calculated using mPLM. Secondly, we connect nodes created by machine translation with their original texts.

3.2 Heterogeneous Graph Convolution

After building the heterogeneous cross-lingual graph, we first encode all the nodes using mPLM by directly inputting the text to the mPLM and taking the hidden states of the first token. The encoded node features are fixed during training. Next we apply heterogeneous graph convolutional network (Hetero-GCN) (Hu et al., 2019) on the graph to calculate higher-order representations of each node with aggregated information.

Heterogeneous GCN applies traditional GCN on different sub-graphs separated by different types of edges and aggregates information to an implicit common space.

$$H^{(l+1)} = \sigma(\sum_{\tau \in \mathcal{T}} \tilde{A}_\tau \cdot H_\tau^{(l)} \cdot W_\tau^{(l)}) \qquad (1)$$

where $\tilde{A}_\tau$ is a submatrix of the symmetric normalized adjacency matrix that only contains edges with type τ, $H_\tau^{(l)}$ is the feature matrix of the neighboring nodes with type τ of each node, and $W_\tau^{(l)}$ is a trainable parameter. $\sigma(\cdot)$ denotes a non-linear activation function, which we use leaky ReLU. Initially, $H_\tau^{(0)}$ is the node feature calculated by mPLM.

Empirically, we use two graph convolution layers to aggregate information within second-order neighbors. Then a linear transformation is applied to the document nodes to get the predictions.

4 Experiments

We evaluate our framework on three different classification tasks, including Amazon Review sentiment classification (Prettenhofer and Stein, 2010), news category classification from XGLUE (Liang et al., 2020), and intent classification on a multilingual spoken language understanding (SLU) dataset (Schuster et al., 2019). More details of each dataset is provided in the appendix. For all the tasks, we use only the English samples for training and evaluate on other 6 languages, which are German (DE), French (FR), Russian (RU), Spanish (ES), Japanese (JA), and Thai (TH).

4.1 Experiment Setting

In all our experiments, we use two-layer GCN with hidden size 512 and output size 768. Each document is connected with 3 most similar documents. The model is trained using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 2×10^{-5} and batch size 256. We train the GCN for at most 15 epochs and evaluate the model with best performance on validation set. XLM-RoBERTa (Conneau et al., 2020) is used to encode all the documents and words, which is finetuned on the English training set of each task for 2 epochs with batch size 32 and learning rate 4×10^{-5}. We set the max length as 128 for intent classification, and 512 for the other two tasks. Each experiment is repeated 3 times and the average accuracy is reported. All the experiments are conducted on an NVIDIA V100 GPU [2].

For part-of-speech tagging, we adopt different taggers for each language [3] and map all the tags to

[1] https://cloud.tencent.com/document/api/551/15619

[2] Our codes are available at https://github.com/TencentGameMate/gnn_cross_lingual.

[3] We choose Stanford POS Tagger (Toutanova et al., 2003) for EN, Spacy (https://spacy.io) for DE, FR and ES, MeCab (https://taku910.github.io/mecab/) for JA, tltk (https://pypi.org/project/tltk/) for TH, and nltk (https://www.nltk.org) for RU.

Method	EN → DE				EN → FR				EN → JA			
	books	dvd	music	avg	books	dvd	music	avg	books	dvd	music	avg
CLDFA	83.95	83.14	79.02	82.04	83.37	82.56	83.31	83.08	77.36	80.52	76.46	78.11
MVEC	88.41	87.32	89.97	88.61	89.08	88.28	88.50	88.62	79.15	77.15	79.70	78.67
mBERT	84.35	82.85	93.85	83.68	84.55	85.85	83.65	84.68	73.35	74.80	76.10	74.75
XLM	86.85	84.20	85.90	85.65	88.10	86.95	86.20	87.08	80.95	79.20	78.02	79.39
XLM-R	91.65	87.60	90.97	90.07	89.33	90.07	89.15	89.52	85.26	86.77	86.95	86.33
CLHG	**92.70***	**88.60***	**91.62***	**90.97***	**90.67***	**91.38***	**90.45***	**90.83***	**87.21***	**87.33***	**88.08***	**87.54***

Table 1: Sentiment classification accuracy (%) on Amazon Review dataset. * shows the result is significantly better than XLM-R baseline with p-value ≤ 0.05.

Method	DE	FR	ES	RU
mBERT	82.6	78.0	81.6	79.0
XLM-R	84.5	78.2	83.2	79.4
Unicoder	84.2	78.5	83.5	79.7
XLM-R (ours)	83.99	78.66	83.27	80.42
CLHG	**85.00**$^{+}$	**79.58***	**84.80***	**80.91**$^{+}$

Table 2: Classification accuracy (%) on XGLUE News Classification. We re-run the XLM-R model and also report our reproduced results. * shows the result is significantly better than XLM-R baseline with p-value ≤ 0.05, and $^{+}$ indicates p-value ≤ 0.1.

Method	EN → ES	EN → TH
CoSDA_ML+mBERT	94.80	76.80
CoSDA_ML+XLM	90.30	86.70
mBERT	74.91	42.97
XLM	62.30	31.60
XLM-R	94.38	85.17
CLHG	**96.81***	**89.71***

Table 3: Intent classification accuracy (%) on multilingual SLU dataset. * shows the result is significantly better than XLM-R baseline with p-value ≤ 0.05.

Universal Dependency (UD) tagset [4].

4.2 Baselines

Our method is compared with different multilingual pretrained models finetuned for each task, which include multilingual BERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019) and XLM-RoBERTa (Conneau et al., 2020), and also with published state-of-the-art results on each dataset.

Amazon Review. CLDFA (Xu and Yang, 2017) utilizes model distillation and parallel corpus to transform a model from source to target language. MVEC (Fei and Li, 2020) refines the shared latent space of mPLM with unsupervised machine translation and a language discriminator.

[4] https://universaldependencies.org

XGLUE News Classification. Unicoder (Huang et al., 2019) is another mPLM proposed recently, and is used as the baseline method provided alongside XGLUE benchmark.

Multilingual SLU. CoSDA-ML (Qin et al., 2020) is a data augmentation framework that automatically generates code-switching documents using a bilingual dictionary, which is used when finetuning language models on downstream tasks.

4.3 Results and Analysis

The results are provided in table 1 2 and 3 for each dataset. Our method significantly outperforms state-of-the-art baselines and achieves consistent improvements over XLM-R model. The most performance gain is achieved on the multilingual SLU dataset. Different from the other two, this dataset consists of short texts, and thus the created graph is much cleaner and more suitable for GCN to model.

To verify that the improvement does not come barely from the external data, we conduct another experiment that adds the translated data to the training set and finetunes the baseline XLM-R model on Amazon Review dataset. The results showed very slight improvement (0.09% on average), showing that XLM-R cannot directly benefit from external training data.

Ablation studies are performed on Amazon Review dataset to analyze the effectiveness of different graph structures. From the results provided in table 4, variant 1 containing a homogeneous graph with only word-doc edges (same as TextGCN) performs the worst, while adding more information leads to better performance in general. Comparing variants 4-7, similarity demonstrates to be the most important among all added information. Similarity edges help the model to converge faster and learn better as well, since they provide a "shortcut" between documents that is highly likely to be

	XLM-R	1	2	3	4	5	6	7	full model
word-doc		✓		✓	✓	✓	✓	✓	✓
POS tags						✓	✓	✓	✓
translation edges			✓		✓		✓	✓	✓
similarity edges			✓	✓	✓	✓		✓	✓
unlabeled			✓	✓	✓	✓	✓		✓
EN → DE	90.01	87.60	90.60	90.75	90.77	90.67	89.90	**91.26**	90.97
EN → FR	89.52	90.62	89.95	90.65	90.70	90.37	89.82	90.85	**90.92**
EN → JA	86.61	86.26	87.19	87.31	87.35	87.44	87.18	86.57	**87.54**

Table 4: Ablation study results. The left-most column shows the results of finetuned XLM-R, and others each indicates a variant in graph construction. We conduct experiments on the Amazon Review dataset and report the average accuracy across three domains.

in the same category. Variant 7 shows that unlabeled corpus play an important role in EN→JA setting, but less effective when transferring between similar languages, since unlabeled data inevitably contain some noise and do not provide much help for linguistically-closer languages. Variant 4 also shows that POS tags are more helpful for distant language pairs like EN→JA, and our added experiment on EN→TH shows greater impact of POS tags (89.71→88.06 when removing POS tags). Additionally, we test a variant without any external tool that requires training resources in the target language. Variant 3 does not rely on POS tagger or translation service, and still outperforms the XLM-R baseline with a large margin. This demonstrates that our method can be adopted for real low-resource languages without good tagger or translator.

5 Conclusion

In this study, we propose a novel graph-based method termed CLHG to capture various kinds of information within and across languages for cross-lingual text classification. Extensive experiments illustrate that our framework effectively extracts and integrates heterogeneous information among multilingual corpus, and these heterogeneous relations can enhance existing models and are instrumental in cross-lingual tasks. There may exist some better semantic or syntactic features and combinations of features, which we leave as a future work to explore. We also wish to extend our GNN-based framework to different NLP tasks requiring knowledge transfer and adaptation in the future.

Acknowledgement

We thank all the anonymous reviewers for their careful reading and helpful comments. We thank our colleagues for their thoughtful and inspiring discussions during the development of this work. We appreciate Tencent Cloud for providing computational resources and technical support.

References

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Andrade, Kunihiko Sadamasa, Akihiro Tamura, and Masaaki Tsuchida. 2015. Cross-lingual text classification using topic-dependent word probabilities. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1466–1471.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 547–556.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936, Online. Association for Computational Linguistics.

Xin Luna Dong and Gerard de Melo. 2019. A robust self-learning framework for cross-lingual text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6307–6311.

Kevin Duh, Akinori Fujino, and Masaaki Nagata. 2011. Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 429–433.

Hongliang Fei and Ping Li. 2020. Cross-lingual unsupervised sentiment classification with multi-view transfer learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5759–5771.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. 2020. Multi-scale self-attention for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7847–7854.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, Hong Kong, China. Association for Computational Linguistics.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Zheng Li, Mukul Kumar, William Headden, Bing Yin, Ying Wei, Yu Zhang, and Qiang Yang. 2020. Learn to cross-lingual transfer with meta graph learning across heterogeneous languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2290–2301, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv*, abs/2004.01401.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1118–1127.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data

augmentation for zero-shot cross-lingual nlp. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3853–3860.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805.

Lei Shi, Rada Mihalcea, and Mingjun Tian. 2010. Cross language text classification by model translation and semi-supervised learning. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1057–1067.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4067–4076, Online. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Ruochen Xu and Yiming Yang. 2017. Cross-lingual distillation for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Vancouver, Canada. Association for Computational Linguistics.

Ruochen Xu, Yiming Yang, Hanxiao Liu, and Andrew Hsi. 2016. Cross-lingual text classification via model translation with limited dictionaries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 95–104, New York, NY, USA. Association for Computing Machinery.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Mozhi Zhang, Yoshinari Fujinuma, and Jordan Boyd-Graber. 2020a. Exploiting cross-lingual subword similarities in low-resource document classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9547–9554.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020b. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 334–339, Online. Association for Computational Linguistics.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1412.

Yftah Ziser and Roi Reichart. 2018. Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 238–249.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.

A Datasets

Here we introduce the three datasets we use in our experiments. A summary of the statistics for three datasets are provided in table 5.

Amazon Review [5] This is a multilingual sentiment classification dataset covering 4 languages (English, German, French, Japanese) on three domains (*Books, DVD*, and *Music*). The original dataset contains ratings of 5-point scale. Following previous works (Xu and Yang, 2017; Fei and Li, 2020), we convert the ratings to binary labels with threshold at 3 points. Since English is not used for testing, we follow the previous works and re-construct the training and validation set by combining the English training and test set. The new training set contains 3,200 randomly sampled documents. We use the training set of target languages for validation. This dataset also provides large amount of unlabeled data for each language, which is used in our framework.

XGLUE News Classification [6] This is a subtask of a recently released cross-lingual benchmark named XGLUE. This subtask aims at classifying the category of a news article, which covers 10 categories in 5 languages, including English, Spanish, French, German and Russian. This dataset does not contain unlabeled data for target languages, so we do not add unlabeled documents in our graph either.

Multilingual SLU [7] This dataset contains short task-oriented utterances in English, Spanish and Thai across weather, alarm and reminder domains. We evaluate our framework on the intent classification subtask, which has 12 intent types in total. For each target language, we use the original training set as unlabeled data added in the graph.

B Hyperparameter Search

We perform a grid search to pick the best combination of hyperparameters. The hidden size and output size are chosen among {384, 512, 768}, and the learning rate within $\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$. For the XLM-R baseline, we also tune the learning rate within $\{2 \times 10^{-5}, 4 \times 10^{-5}, 5 \times 10^{-5}\}$ and number of epochs from 2 to 5. Among all the combination of hyperparameters, we pick the values with the best performance on the German training set from *Books* domain of Amazon Review dataset, and use the same set of hyperparameters for all our experiments. The maximum length of XLM-R model is chosen based on statistics of the datasets.

[5] https://webis.de/data/webis-cls-10.html

[6] https://microsoft.github.io/XGLUE/

[7] https://fb.me/multilingual_task_oriented_data

dataset	#category	language	#train	#valid	#test	#unlabeled	avg. length
Amazon Review (Books)	2	English	2,000	/	2,000	50,000	168.31
		German	2,000	/	2,000	165,457	151.27
		French	2,000	/	2,000	32,868	123.84
		Japanese	2,000	/	2,000	169,756	155.05
Amazon Review (Dvd)	2	English	2,000	/	2,000	30,000	167.31
		German	2,000	/	2,000	91,506	158.58
		French	2,000	/	2,000	9,356	138.89
		Japanese	2,000	/	2,000	68,324	150.87
Amazon Review (Music)	2	English	2,000	/	2,000	25,220	146.18
		German	2,000	/	2,000	60,382	143.50
		French	2,000	/	2,000	15,940	142.21
		Japanese	2,000	/	2,000	55,887	131.62
XGLUE NC	10	English	100,000	10,000	10,000	/	553.65
		German	/	10,000	10,000	/	484.69
		French	/	10,000	10,000	/	567.86
		Spanish	/	10,000	10,000	/	533.09
		Russian	/	10,000	10,000	/	426.80
Multilingual SLU	12	English	30,521	4,181	8,621	/	8.05
		Spanish	3,617	1,983	3,043	/	8.74
		Thai	2,156	1,235	1,692	/	5.12

Table 5: Summary statistics of the datasets. The average length shows the average number of words in each document.

Towards More Equitable Question Answering Systems: How Much More Data Do You Need?

Arnab Debnath*, **Navid Rajabi***, **Fardina Fathmiul Alam***, **Antonios Anastasopoulos**
Department of Computer Science, George Mason University
{adebnath,nrajabi,falam5,antonis}@gmu.edu

Abstract

Question answering (QA) in English has been widely explored, but multilingual datasets are relatively new, with several methods attempting to bridge the gap between high- and low-resourced languages using data augmentation through translation and cross-lingual transfer. In this project, we take a step back and study which approaches allow us to take the most advantage of *existing* resources in order to produce QA systems in *many* languages. Specifically, we perform extensive analysis to measure the efficacy of few-shot approaches augmented with automatic translations and permutations of context-question-answer pairs. In addition, we make suggestions for future dataset development efforts that make better use of a fixed annotation budget, with a goal of increasing the language coverage of QA datasets and systems.[1]

1 Introduction

Automatic question answering (QA) systems are showing increasing promise that they can fulfil the information needs of everyday users, via information seeking interactions with virtual assistants. The research community, having realized the obvious needs and potential positive impact, has produced several datasets on information seeking QA. The effort initially focused solely on English, with datasets like WikiQA (Yang et al., 2015), MS MARCO (Nguyen et al., 2016), SQuAD (Rajpurkar et al., 2016), QuAC (Choi et al., 2018), CoQA (Reddy et al., 2019), and Natural Questions (NQ) (Kwiatkowski et al., 2019), among others. More recently, heading calls for linguistic and typological diversity in natural language processing research (Joshi et al., 2020), larger efforts have produced datasets in multiple languages, such as TyDi QA (Clark et al., 2020), XQuAD (Artetxe et al., 2020), or MLQA (Lewis et al., 2020).

Despite these efforts, the linguistic and typological coverage of question answering datasets is far behind the world's diversity. For example, while TyDi QA includes 11 languages –less than 0.2% of the world's approximately 6,500 languages (Hammarström, 2015)– from 9 language families, its typological diversity is 0.41, evaluated in a [0,1] range with the measure defined by Ponti et al. (2020); MLQA provides data in 7 languages from 4 families, for a typological diversity of 0.32. The total population coverage of TyDi QA, based on population estimates from Glottolog (Nordhoff and Hammarström, 2012), is less than 20% of the world's population (the TyDiQA languages total around 1.45 billion speakers).

Obviously, the ideal solution to this issue would be to collect enough data in every language. Unfortunately, this ideal seems unattainable at the moment. In this work, we perform extensive analysis to investigate the next-best solution: using the existing resources, large multilingual pre-trained models, data augmentation, and cross-lingual learning to improve performance with just a few or no training examples. Specifically:

- we study how much worse a multilingual few-shot training setting would perform compared to training on large training datasets,
- we show how data augmentation through translation can reduce the performance gap for few-shot setting, and
- we study the effect of different fixed-budget allocation for training data creation across languages, making suggestions for future dataset creators.

[1]Code and data for reproducing our experiments are available here: https://github.com/NavidRajabi/EMQA.

*Equal contribution.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 621–629
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Problem Description and Settings

We focus on the task of simplified minimal answer span selection over a **gold passage**: The inputs to the model include the full text of an article (the *passage* or *context*) and the text of a question (*query*). The goal is to return the start and end byte indices of the minimal span that completely answers the question.

Our models follow the current state-of-the-art in extractive question answering, relying on large multilingually pre-trained language models (in our case, multilingual BERT (Devlin et al., 2019)) and the task-tuning strategy of Alberti et al. (2019), which outperforms approaches like DocumentQA (Clark and Gardner, 2018) or decomposable attention (Parikh et al., 2016). In all cases, we treat the official TyDi QA development set as our test set, since the official test set is not public.[2] We provide concrete details (model cards, hyperparmeters, etc) on our model and training/finetuning regime in Appendix A.

To simulate the scenario of data-scarce adaptation of such a model to unseen languages, we will treat the TyDi QA languages as our test, unseen ones. We will assume that we have access to (a) other QA datasets in more resource-rich languages (in particular, the SQuAD dataset which provides training data in English), and (b) translation models between the languages of existing datasets (again, English) and our target "unseen" languages.

In the experiments sections, we first focus on few- and zero-shot experiments (§3) and then study the effects of language selection and budget-restricted decisions on training data creation (§4).

Evaluation We report F1 score on the test set of each language, as well as a macro-average excluding English ($\mathbf{avg}_{\mathcal{L}}$). In addition, to measure the expected impact on actual systems' users, we follow Faisal et al. (2021) in computing a population-weighted macro-average ($\mathbf{avg}_{\mathbf{pop}}$) based on language community populations provided by Ethnologue (Eberhard et al., 2019).

3 Is Few-Shot a Viable Solution?

We first set out to explore the effect of the amount of available data on downstream performance. Starting with baselines relying solely on English-only SQuAD, we implement a few-shot setting for fine-tuning on the target languages of TyDi QA.[3] To our knowledge, this is the first study of its type on the TyDi QA benchmark.

The straightforward baseline simply provides zero-shot results on TyDi QA after training only on English. Table 1 provides our (improved) reproduction of the baseline experiments of Clark et al. (2020). The skyline results (bottom of Table 1) reflect the presumably best possible results under our current modeling approach, which trains jointly on all languages using all available TyDi QA training data. We note that for most languages the gap between the baseline and the skyline is more than 20 percentage points, with the exception of English where –unsurprisingly– there is a difference of only 3.3 percentage points. The performance gap is smallest for Russian (rus) at 10.9 percentage points, and largest for Telugu (tel) at 34 points.

We first study a *monolingual* few-shot setting. That is, we fine-tune the model trained on the English SQuAD dataset, with only a small amount of data (10, 20, or 50 training instances) in the test language. Due to space limitations, we only present results with 50 examples per language in Table 1, but the full experiments are available in Appendix C. We observe that even just 50 additional training instances are enough for significant improvements, which are consistent across all languages. For example, the improvement in Finnish (fin) exceeds 15 percentage points and covers about more than 60% of the performance gap between the baseline and the skyline.

We now turn to a *multilingual* few-shot setting. Exactly as before, we assume a scenario where we only have access to a small amount of data in each language, but now we fine-tune using that small amount of data in all languages. For example, 10 training instances in each language result in training with 90 training examples over the 9 test languages. A sample of our experimental results are presented in Table 1 under "multilingual few-shot," with complete results in Appendix C.

Simply adding 50 instances from each language we obtain an F1 score of 67.9 over the zero-shot baseline, an improvement of almost 7 percentage points which reduces the zero-full gap by 43.4%.

[2] This follows the guidelines to perform analyses over the development set to ensure the integrity of the leaderboard.

[3] We do not report results on Korean, due to a late-discovered issue: we found that parts of the Korean data use a Unicode normalization scheme different than what is expected by mBERT's vocabulary. We suspect this is responsible for our Korean results being consistently around 50% worse than previously published results.

Model	Results (F1-score) eng	ara	ben	fin	ind	swa	rus	tel	$\text{avg}_{\mathcal{L}}$ (without eng)	avg_{pop}
Baseline: SQuAD zero-shot										
(reproduction)	74.2	59.0	57.3	55.7	63.2	60.3	65.6	44.6	58.0±6.3	59.3
Monolingual Few-Shot (+50)	73.9	64.9	66.4	70.9	73.3	70.1	66.3	62.5	67.8±3.5	67.1
Multilingual Few-Shot										
(+10/lang, 90 total)	73.7	64.6	62.9	66.5	67.0	63.1	65.9	59.6	64.2±2.4	64.4
(+50/lang, 450 total)	73.4	69.2	65.8	69.0	73.4	68.8	67.2	66.2	68.5±2.4	68.6
(+100/lang, 900 total)	74.2	72.5	70.9	71.9	75.5	72.3	69.3	69.3	71.7±2.0	71.9
(+500/lang, 4500 total)	76.1	76.3	74.5	78.2	81.4	79.2	73.3	73.7	76.7±2.8	76.2
Data Augmentation + Multilingual Few-Shot										
+tSQuAD	74.9	65.4	58.4	66.7	65.2	69.4	60.2	44.7	61.4±7.7	61.2
+mSQuAD	75.1	65.6	68.6	71.7	70.3	66.2	75.5	49.4	66.7±7.7	67.6
+mSQuAD +500/lang	77.6	78.7	75.0	78.5	83.5	82.5	73.2	75.3	78.1±3.6	77.6
+tSQuAD +500/lang	77.9	78.8	80.0	79.5	82.8	83.6	72.5	73.5	78.7±3.9	78.6
Skyline: Full training on TyDi QA train										
(reproduction)	*77.5*	*82.4*	*78.9*	*80.1*	*85.4*	*83.8*	*76.5*	*78.3*	*80.8±3.0*	*80.9*

Table 1: Data augmentation combined with multilingual few-shot learning can reach about 98% of the skyline accuracy using only 10 times less training data on the test languages beyond English.

We note that the total 450 training instances represent less than 1% of the full TyDi QA training set! Doubling that amount of data to 100 examples per language further increases downstream performance to an average overall F1 score of 71.7. Going further to the point of adding 500 training instances per language (for a total of 4500 examples) leads to even larger improvements for an average F1 score of 76.7. That is, using less than 10% of the available training data we can reduce the average F1 score performance gap by more than 82%. For a few languages the gap reduction is even more notable, e.g., more than 92% for Finnish.

Data Augmentation through Translation Generating translations of English dataset to train systems in other languages has a long history and has been successful in the QA context as well (Yarowsky et al., 2001; Xue et al., 2020, *inter alia*). We follow the same approach, translating all SQuAD paragraphs, questions, and answers to all TyDi QA languages using Google Translate.[4] For each language, we keep between 20-50% of the question-answer pairs where the translated answer has an exact match in the translated paragraph, which becomes the target span.[5] Details of the resulting dataset (which we refer to as tSQuAD) are in Table 3 in Appendix B. A second approach translates the question of a training instance into one language, but keeps the answer and context into the original language. The result is a modified training set (which we name mSQuAD) that requires better cross-lingual modeling, as the question and contexts are in different languages.

Both approaches improve over the zero-shot baseline with F1 score of 61.4 (+3) and 66.7 (+8). Notably, though, they are not as effective as few-shot training even with just 50 instances per languages. This further strengthens the discussion of Clark et al. (2020) on the qualitative differences between the SQuAD and TyDi QA dataset. Nevertheless, combining tSQuAD (or mSQuAD) with a few examples from the TyDi QA dataset leads to our best-performing methods. In particular, augmentation through translation leads to an 1-2 percentage point improvements over the multilingual few-shot approach (cf. 76.7 to 78.1/78.7 F1 score in Table 1; full results in Appendix C). Now, using only 500 new training examples per language we are *almost* (98%) at similar performance levels as the skyline.

[4]We release the data to facilitate the reproduction of our experiments.

[5]This approach could be enhanced using word/phrase alignment techniques, which we leave for future work.

Results (F1-score)								Overall	Δ_l	avg	
eng	ara	ben	fin	ind	swa	rus	tel	(w/o eng)	(max-min)	seen	unseen
Baseline: no budget for additional data (zero-shot except for eng)											
74.2	59.0	57.3	55.7	63.2	60.3	65.6	44.6	58.0±6.3	29.6	74.2	58.0
Monolingual budget allocation (max 4500 per language; 7 experiments)											
76.0±1.8	74.0±3.9	69.1±5.0	75.8±2.7	78.4±4.1	71.7±4.1	75.7±6.3	61.3±12.3	72.3±5.3	17.1	77.1	71.3
Tri-lingual budget allocation (1500 per language; 7 random language selection experiments)											
76.7±1.2	77.2±2.8	68.6±4.8	77.9±1.6	80.9±3.3	81.5±3.3	72.7±2.3	62.9±13.3	74.5±6.3	18.6	78.9	68.5
Uniform budget allocation (500 per language)											
77.9	78.8	80.0	79.5	82.8	83.6	72.5	73.5	78.7±3.9	11.1	78.6	–
Ideal Few-Shot (4500 in each language; in-language results)											
78.4	*81.8*	*77.7*	*79.7*	*83.9*	*84.0*	*75.7*	*78.2*	*79.9±3.0*	*8.3*	*79.9*	-

Table 2: A more egalitarian budget allocation leads to better *and* more equitable performance across languages (avg±std: higher average, lower std. deviation) reducing the gap (Δ_l) between best and worst performing languages.

4 How to Spend the Annotation Budget?

In the previous section we show that the combination of data augmentation techniques with a few new annotations can reach almost 98% of the performance one would obtain by training on 10x more data. In this section we explore how one should allocate a fixed annotation budget, in order to achieve not only higher average but also more *equitable* performance across languages.

Keeping our budget fixed to 4500 instances, we study 3 scenarios. The first is **monolingual** allocation, where the whole budget is consumed by collecting training examples on a single language. We repeat the study over all 8 languages of our test set, randomly sampling training instances from the TyDi QA training set. Second, we study a **tri-lingual** budget allocation scheme, where we equally split the budget across 3 languages for 1500 training instances per language. We repeat this experiment 7 times, each time randomly selecting 3 languages. Last, the third and more **egalitarian** scenario splits the budget equally across all 8 languages, matching our previously analyzed few-shot scenario where we only have 500 additional training examples per language. In all experiments, we use our best-performing approach from the previous section, also utilizing tSQuAD for pre-training.

Our findings are summarized in Table 2. For the repeated monolingual and tri-lingual scenarios we report average performance across our experiment repetitions (full results in Appendix E). We can conclusively claim that a uniform budget allocation leads to not only better average performance, but also to more equitable performance. We report two straightforward measures for the equitability of the average accuracy across languages. First, we report the standard deviation of the accuracy across languages; the lower the standard deviation, the more equitable the performance. We also report the difference between the best and the worst performing language for each experiment, as well as the averages for the languages that are seen and unseen during fine-tuning.

Having no budget for additional annotation (essentially, attempting the task in zero-shot fashion) leads to the most inequitable performance. The monolingual scenario typically leads to the highest accuracy when evaluating on the same language as the new training examples (the *ideal* section of Table 2) but the zero-shot performance on all other languages is generally significantly worse, leading to inequity. The tri-lingual scenarios follow similar patterns, with performance close to state-of-the-art for the four languages (three plus English) that have been included in the fine-tuning process, but with the rest of the languages lagging behind: the difference between seen and unseen languages is on average 10.4 points. In our experiments we randomly sampled (without replacement) three of the seven languages, but one could potentially use heuristics or a meta-model like that of Xia et al. (2020) to find or suggest the best subset of candidate languages for transfer learning; we leave such an investigation for future work.

Encouragingly, the uniform budget allocation scenario leads to higher average performance, while also reducing the gap between worst and best performing languages from around 30 percentage points to less than 12 points (60% reduction). Note that a 8x larger budget (*ideal* scenario) with 4500 instances per language would further improve downstream accuracy and equitability. Note that in this case where some resources are available,

simple multilingual fine-tuning might not be the best approach for some languages, e.g. compared to monolingual fine-tuning or meta-learning approaches (Wang et al., 2020; Muller et al., 2021, *inter alia*). We leave an investigation of such settings for future work.

5 Discussion

We show that data augmentation through translation along with few-shot fine-tuning on new languages with a uniform budget allocation leads to a performance close to 98% of an approach using 10x more data, while producing more equitable models than other budget-constrained alternatives.

The implications of our findings become clear with a counter-factual exploration. The Gold Passage portion of the TyDi QA dataset includes around 87,000 annotated examples (50k for training across 9 languages and about 37k development and test samples). Consider the scenario where, given this annotation budget, we maintain the same evaluation standards collecting 4k development and test examples per language, but we only collect 500 training examples per language. In that case, we could have created a much more diverse resource that would include at least 19 languages! Now consider the expectation of the downstream accuracy in our counterfactual scenario: uniform budget allocation on 19 languages would lead to an average accuracy (F1 score) of around 78% (similar to our experiments). Instead, under the (currently factual) scenario where we only have training data for 9 languages, the average accuracy for these 9 languages is around 80%, but the zero-shot expected average on the other 10 languages is 10 points worse – in that case, the overall average accuracy would be around 74%, 4 points lower than that of the egalitarian allocation scenario. Hence, as long as the ideal scenario of collecting a lot of data for a lot of languages remains infeasible, we suggest that the community puts an additional focus on the linguistic diversity of our evaluation sets and use other techniques to address the lack of training data.

Acknowledgements

This work is supported by NSF Award 2040926. The authors also want to thank Fahim Faisal for helpful discussions on setting up the experiments. Most experiments were run on ARGO,[6] a research computing cluster provided by the Office of Research Computing at George Mason University, VA, and a few experiments were run on Amazon Web Services instances donated through the AWS Educate program.

[6] http://orc.gmu.edu

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A BERT baseline for the natural questions. arXiv:1901.08634.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184. Association for Computational Linguistics.

Christopher Clark and Matt Gardner. 2018. Simple and Effective Multi-Paragraph Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol 1*, pages 4171–4186. Association for Computational Linguistics.

David M Eberhard, Gary F Simons, and Charles D. (eds.) Fennig. 2019. Ethnologue: Languages of the world. 2019. *online. Dallas, Texas: SIL International.*

Fahim Faisal, Sharlina Keshava, Md Mahfuz ibn Alam, and Antonios Anastasopoulos. 2021. SD-QA: Spoken Dialectal Question Answering for the Real World. Preprint.

Harald Hammarström. 2015. " ethnologue" 16/17/18th editions: A comprehensive review. *Language*, pages 723–737.

Hugging Face - mBERT. 2020. Hugging Face - bert-base-multilingual-cased. [Online; accessed 01-Novemberr-2020].

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Sebastian Nordhoff and Harald Hammarström. 2012. Glottolog/Langdoc:Increasing the visibility of grey literature for low-density languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3289–3294. European Language Resources Association (ELRA).

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. arXiv:2010.11934.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

A Experimental Settings

For the experiments, we've used "`bert-multi-lingual-base-uncased`" (mBERT) (Hugging Face - mBERT, 2020) as mentioned as the main baseline on TyDi QA paper (Clark et al., 2020). It is a pre-trained model on the top 102 languages with the largest Wikipedia using a masked language modeling (MLM) objective (Devlin et al., 2019). From preliminary experiments, we realized that the optimum trade-off between the highest F1 score and the least computational cost is achieved by training for 3 epochs, using batch size of 24, and learning rate of 3e-5. Therefore, we applied these hyperparameter settings for our experiments. The main script we used was a module under the Huggingface library (Wolf et al., 2020) (called run_squad), which is being used widely for fine-tuning transformers for multi-lingual question answering datasets.

B SQuAD Translation Details

We augmented the English SQuAD with *translated SQuAD* (tSQuAD) instances for each language. Here, the contexts, questions and answers from SQuAD instances are translated to the target languages using Google Translate (with the `google-trans-new` API) and only the instances where an exact match of translated answer is found in the translated context, are kept for augmentation. The total number of instances per language, we ended up with after translation is listed in Table 3.

C Complete Few-Shot Experiments

Provided in Table 4.

D Mix-and-Match Experiments

Provided in Table 5.

E Budget Allocation Experiments

The complete results for our experiments are presented in Table 6.

	SQuAD	tAr	tBn	tFin	tInd	tKo	tRus	tSwa	tTel
no of paragraphs	18.9	16.6	13.5	12.4	16.2	11.2	11.6	15.3	16.6
no of QAs	87.6	39.1	24.1	21.4	36.1	18.1	19.2	31.2	39.7

Table 3: Number (in 1000s) of paragraphs and QA pairs present in the original SQuAD and translated SQuAD

Model	**Results (F1-score)** eng	ara	ben	fin	ind	swa	rus	tel	**Overall** (without eng)
Baseline: SQuAD zero-shot									
(Clark et al., 2020)	73.4	60.3	57.3	56.2	60.8	52.9	64.4	49.3	57.3±4.7
(ours)	74.2	59.0	57.3	55.7	63.2	60.3	65.6	44.6	58.0±6.3
Monolingual Few-Shot (+10)	73.7	64.7	62.8	68.2	69.3	59.9	65.6	50.7	63.0±5.8
Monolingual Few-Shot (+20)	74.7	63.5	60.5	66.6	72.1	63.9	66.8	63.0	65.2±3.4
Monolingual Few-Shot (+50)	73.9	64.9	66.4	70.9	73.3	70.1	66.3	62.5	67.8±3.5
Multilingual Few-Shot									
(+10/lang, 90 total)	73.7	64.6	62.9	66.5	67.0	63.1	65.9	59.6	64.2±2.4
(+20/lang, 180 total)	73.9	65.9	66.8	69.0	72.5	64.2	66.9	63.7	67.0±2.8
(+50/lang, 450 total)	73.4	69.2	65.8	69.0	73.4	68.8	67.2	66.2	68.5±2.4
(+100/lang, 900 total)	74.2	72.5	70.9	71.9	75.5	72.3	69.3	69.3	71.7±2.0
(+200/lang, 1800 total)	73.9	74.8	70.5	74.1	77.7	76.4	69.8	70.0	73.3±3.0
(+500/lang, 4500 total)	76.1	76.3	74.5	78.2	81.4	79.2	73.3	73.7	76.7±2.8
Data Augmentation + Multilingual Few-Shot									
+tSQuAD(50/lang)	73.8	64.0	62.4	68.4	69.7	59.7	66.8	48.1	62.7±6.8
+tSQuAD(100/lang)	72.4	62.2	66.6	68.4	68.6	64.9	67.1	47.5	63.6±6.9
+tSQuAD(200/lang)	74.4	62.7	64.2	68.8	70.7	66.1	66.2	48.3	63.9±6.8
+tSQuAD(500/lang)	73.7	63.2	69.5	67.9	70.9	69.8	66.7	49.1	65.3±7.0
+tSQuAD(all)	74.9	65.4	58.4	66.7	65.2	69.4	60.2	44.7	61.4±7.7
+mSQuAD +500/lang	77.6	78.7	75.0	78.5	83.5	82.5	73.2	75.3	78.1±3.6
+tSQuAD +500/lang (mBERT)	77.9	78.8	80.0	79.5	82.8	83.6	72.5	73.5	78.7±3.9
+tSQuAD +500/lang (XLM-R)*	73.2	72.8	78.3	78.5	84.7	80.3	75.0	78.1	78.2±3.5
Skyline: Full training on TyDi QA train									
(Clark et al., 2020)	76.8	81.7	75.4	79.4	84.8	81.9	76.2	83.3	80.4±3.3
(ours)	77.5	82.4	78.9	80.1	85.4	83.8	76.5	78.3	80.8±3.0

Table 4: Complete few-shot and data augmentation results. ∗: Results with XLM-Roberta-Large (Conneau et al., 2020) are generally worse than using mBERT so all other experiments use mBERT.

	Change language of Question only			**Change all; Context & answers the same**		
	Modified Squad	Squad + Modified Squad	Squad + Modified Squad + 500 instances	Modified Squad	Squad + Modified Squad	Squad + Modified Squad + 500 instances
English	66.59	75.06	77.56	65.40	73.49	78.21
Arabic	62.17	65.62	78.70	60.51	65.98	77.96
Bengali	67.33	68.55	75.00	58.60	62.44	76.16
Finnish	67.42	71.67	78.55	62.98	67.58	79.51
Indonesian	66.45	70.33	83.46	61.89	66.44	84.10
Kiswahili	70.32	75.48	82.51	62.66	68.55	80.01
Russian	64.71	66.16	73.16	61.01	65.64	73.28
Telugu	48.32	49.36	75.28	43.62	51.81	74.95
Avg	63.82	66.74	**78.09**	58.76	64.07	**78.00**
SD	6.74	7.75	3.60	6.33	5.31	3.35

Table 5: Mix-and-Match scheme detailed results.

	Results (F1-score)								Overall	Avg	
	eng	ara	ben	fin	ind	swa	rus	tel	(w/o eng)	seen	unseen
Baseline: no budget for additional data (zero-shot excelt in eng)											
	74.2	59.0	57.3	55.7	63.2	60.3	65.6	44.6	60.0±8.5	74.2	58.0
Monolingual budget allocation (max 4500 per language; 7 experiments)											
Arabic	78.4	81.8	62.0	77.6	79.2	72.8	68.0	50.5	70.2±10.3	80.1	68.4
Bengali	74.4	66.3	77.7	71.6	72.8	78.1	66.5	52.0	69.3±8.3	76.1	67.9
Finnish	77.9	75.5	72.6	79.7	81.0	70.6	78.5	52.2	72.9±9.1	78.8	71.7
Indonesian	76.8	76.7	67.4	77.0	83.9	70.2	77.3	52.2	72.1±9.5	80.4	70.1
Kiswahili	76.4	72.5	67.1	75.0	77.4	66.4	84.0	75.0	73.9±5.6	71.4	75.2
Russian	75.2	74.5	66.7	76.3	81.0	75.7	78.8	69.4	74.6±4.7	77.0	73.9
Telugu	73.4	70.6	70.2	73.6	73.7	68.1	77.1	78.2	73.1±3.4	75.8	72.2
	76.0±1.8	74.0±3.9	69.1±5.0	75.8±2.7	78.4±4.1	71.7±4.1	75.7±6.3	61.3±12.3	72.3±5.3	77.1	71.3
Tri-lingual budget allocation (1500 per language; 7 random language selection experiments)											
ben-rus-tel	75.8	72.2	79.0	75.6	74.8	77.1	74.5	76.8	75.7±2.0	76.5	74.9
tel-ind-swa	76.1	75.7	65.5	76.7	83.2	84.7	71.2	77.2	76.3±6.1	80.3	72.3
fin-rus-swa	78.5	76.4	66.3	79.6	80.3	84.8	74.9	53.4	73.7±9.8	79.5	69.1
ara-rus-tel	75.7	79.3	66.8	78.0	79.2	79.9	74.3	77.0	76.4±4.3	76.6	60.8
ara-rus-fin	76.5	80.5	68.9	79.2	80.6	77.5	74.3	53.6	73.5±9.0	77.6	70.2
swa-ind-fin	76.1	77.2	68.5	79.7	84.2	83.0	71.2	51.5	73.6±10.5	80.8	67.1
ara-ind-swa	78.3	79.5	65.4	76.8	83.9	83.5	68.9	50.6	72.7±11.1	81.3	65.4
	76.7±1.2	77.2±2.8	68.6±4.8	77.9±1.6	80.9±3.3	81.5±3.3	72.7±2.3	62.9±13.3	74.5±6.3	78.9	68.5
Uniform budget allocation (500 per language)											
	77.9	78.8	80.0	79.5	82.8	83.6	72.5	73.5	78.7±3.9	78.6	-

Table 6: Complete budget allocation experiments.

Embedding Time Differences in Context-sensitive Neural Networks for Learning Time to Event

Nazanin Dehghani[*1], **Hassan Hajipoor**[*2], **Hadi Amiri**[2,3]
[1]IRISA, University of Rennes 1 (ENSSAT), Lannion, France
[2]Department Computer Science, University of Massachusetts, Lowell, USA
[3]Department of Biomedical Informatics, Harvard University, Boston, USA
nazanin.dehghani@irisa.fr, hassan_hajipoor@student.uml.edu,
hadi_amiri@uml.edu

Abstract

We propose an effective context-sensitive neural model for the task of *time to event* (TTE) prediction, which aims to predict the amount of time to/from the occurrence of given events in streaming content. We investigate this problem in the context of a multi-task learning framework, which we enrich with *time difference* embeddings. To conduct this research, we develop a multi-genre dataset of English events about soccer competitions and academy awards ceremonies, as well as their relevant tweets obtained from Twitter. Our model is 1.4 and 3.3 hours more accurate than the current state-of-the-art model in estimating TTE on English and Dutch tweets respectively. We examine different aspects of our model to illustrate its source of improvement.[1]

1 Introduction

The task of time to event (TTE) prediction aims to determine the amount of time to/from the occurrence of a well-defined event. Accurate prediction of this information is important for temporal tasks such as timeline generation (Reimers et al., 2018), news summarization (Born et al., 2020; Huang et al., 2016), and disease onset prediction in medical domain (Zeliger, 2016; Langbehn et al., 2004).

Current approaches mainly focus on news articles and expect at least one temporal expressions in each input data to predict TTE (Chambers et al., 2014; Reimers et al., 2016, 2018; Hürriyetoğlu et al., 2018; Zhou et al., 2020). These approaches cannot be readily applied to streaming content (such as Twitter data) because such data often do not carry any temporal expressions. Figure 1 show

Figure 1: Examples of tweets that don't carry any explicit time expression but indicate a future or past event due to the implicit temporal connotation in "*looking forward to*," "*must be fun to watch*," "*well done*" etc.

two examples of such tweets.[2] In addition, event-related content in data streams are heavily skewed in time distribution as they are often posted in close proximity of their corresponding events.

The above challenges and intuitions inspire our work to develop a context-sensitive model to predict TTE in streaming content. Our approach is a multi-task learning framework that uses a small fraction of temporally-rich neighbors of each input (tweet) and their time differences (learned through *time difference* embeddings) to predict (a): if the tweet has been posted *before*, *at the same time* or *after* the event, and (b): estimate the absolute value of TTE (in hours) with respect to the tweet. We learn time difference embeddings through an effective character-level sequence to sequence model that takes as input two timestamps and predicts the temporal difference between them (in hours).

The contributions of this paper are as follows: (a) an effective multi-task and context-sensitive framework that uses temporally-rich context and time difference embeddings to accurately predict TTE in streaming content, (b) publicizing a time to event dataset that includes different genres of

[*]First and second authors equally contributed to this work.
[1]Our code and data are available at https://github.com/hajipoor/time2event
[2]In fact, 89% of event-related tweets in our dataset do not carry any time expression.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 630–636
August 1–6, 2021. ©2021 Association for Computational Linguistics

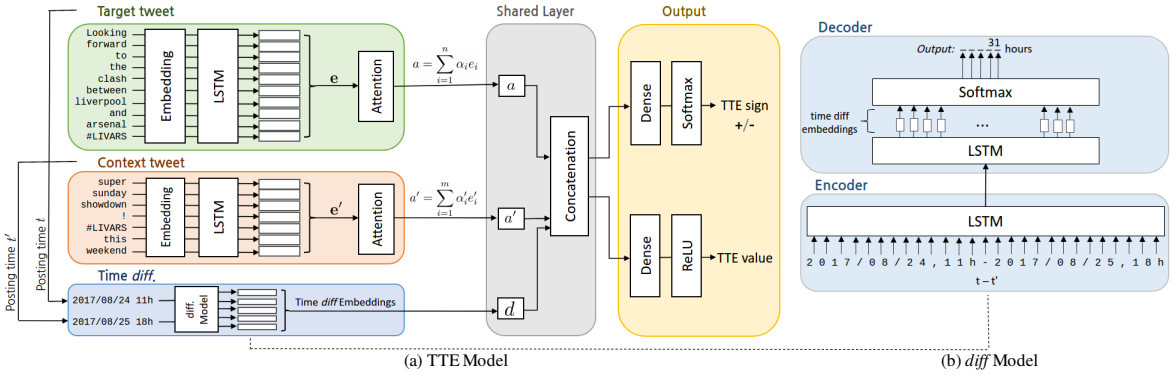


Figure 2: (a) TTE model gives the target tweet and a temporally-rich context tweet, and their time difference embedding learned through *diff* model as input and learns TTE as a combination of regression (TTE value) and classification (TTE sign) tasks. It establishes a common scale between corresponding loss values for effective training and (b) *diff* model gives two times and learns time difference embeddings via sequence to sequence model, which are used in our TTE model.

events (soccer competitions and academy awards ceremonies), their time of occurrence, their relevant tweets as well as TTE information for each tweet.

Our framework is 1.4 and 3.3 hours more accurate than the current state-of-the-art model in estimating TTE on large-scale English and Dutch tweets respectively. In addition, our time difference model achieves an accuracy of 98.3% in terms of creating embeddings that encode temporal differences between given time pairs.

2 Context-sensitive Model

Existing models often assume input data carry explicit temporal information about target events. Although informative, these information may not be available in most textual content, especially in microblogs. We propose to utilize context information (in the form of neighboring tweets) and the relative temporal differences against neighbours to estimate time to event (TTE) for given input texts.

In particular, given a tweet about an event, we propose a multi-task learning framework to predict the absolute value of TTE (in hours) for the tweet, as well as a binary sign which determines if the tweet has been posted before '(+),' or at the same time or after '(-)' the event. Figure 2(a) shows our model for predicting TTE for the target tweet t_i, given its context tweet[3], e.g. a previously posted tweet about the same event, $t_j, j < i$, and their *time difference* embeddings, which encode the time differences between tweet creation times. Our intuition for developing such embeddings is that if context tweets carry useful temporal information about events, then knowing the time differences among tweets could help the model to make more accurate prediction of TTE for the target tweet.

Our model takes as input the concatenation of attention-weighted average embeddings of the target and context tweets (a and a' in Figure 2) and their time difference embedding (d) (see section 2.1). The resulting concatenation are then used to predict *TTE sign* and *TTE value* for the target input. TTE value is a regression task while TTE sign is a binary classification task. To prevent the loss with larger gradient magnitudes dominate the training, we establish a common scale for the different loss magnitudes across the two tasks using the approach proposed in (Kendall et al., 2018), which simultaneously learns classification and regression losses of varying quantities and combines them using homoscedastic uncertainty.

2.1 Time Difference Embeddings

Motivated by recent research on neural numeracy learning (Chen et al., 2019; Wallace et al., 2019), we learn time difference embeddings–`diff embeddings`–as follows: we develop an LSTM-based character-level sequence to sequence model (based on the model presented in (Sutskever et al., 2014)) that takes as input a time pair (t and t') and predicts the difference between them (in hours). The final layer of the model is of size five, where five is determined by the maximum number of digits in the differences of any two timestamps within a 2 years period (i.e., 17520 hours). The final hidden representations of the resulting digits are then concatenated to obtain the diff embeddings, see

[3]Neighboring or context tweets are randomly sampled from the set of previously posted tweets relevant to the target event. Our model can be extended to greater context sizes.

Figure 2(b).

3 Experiments

3.1 Datasets

We develop a dataset from tweets about soccer competitions of the England Premier League (EPL) following the same approach in (Hurriyetoglu et al., 2014; Hürriyetoğlu et al., 2018). We carefully create a list of 42 distinctive hashtags for competitions between seven most famous teams [4]. These matches have the advantage that users tweet about them with distinctive hashtags by convention. We collect tweets that are sent within 14 days of match days between seven popular teams, and obtain the actual time of each event from the EPL schedule.[5]

For the regression task, the tweet label would be the absolute value of the actual time (in hours) to the start of the corresponding event. For the classification task, tweets are labeled as 'before' or 'after' depending on their time of creation against corresponding matches. Our dataset is randomly divided into 80%, 10% and 10% according to events, which are used for training, testing and validation respectively. To study the effect of temporally-rich context in a controlled situation, we divide our dataset of tweets (**All set**) into two disjoint subsets: tweets that carry at least one temporal expression (**T set**), and tweets that have no temporal expression (**N set**), where we use HeidelTime's colloquial temporal tagger (Strötgen and Gertz, 2012, 2013) to extract temporal expressions. We then introduce six new subsets of our data in X-Y format, where $X \in \{$T set, N set, All set$\}$ refers to the type of target tweets and $Y \in \{$T set, N set$\}$ refers to the type of contexts.

To investigate the generalizability of our model on other events, we evaluate our trained model on tweets about the 2018 Academy Awards ceremony. We collected 3K tweets using `#oscars`, `#oscar` and `#academyawards` hashtags in the window of 7 days before and after the date of Oscars 2018. We also use the Dutch dataset to compare our model against the baseline model proposed in (Hurriyetoglu et al., 2014) that developed a hybrid of machine learning and rule-based approach for estimating time to events.

[4]Liverpool, Manchester United, Chelsea, Arsenal, Manchester City, Newcastle United and Tottenham Hotspur

[5]https://www.premierleague.com/

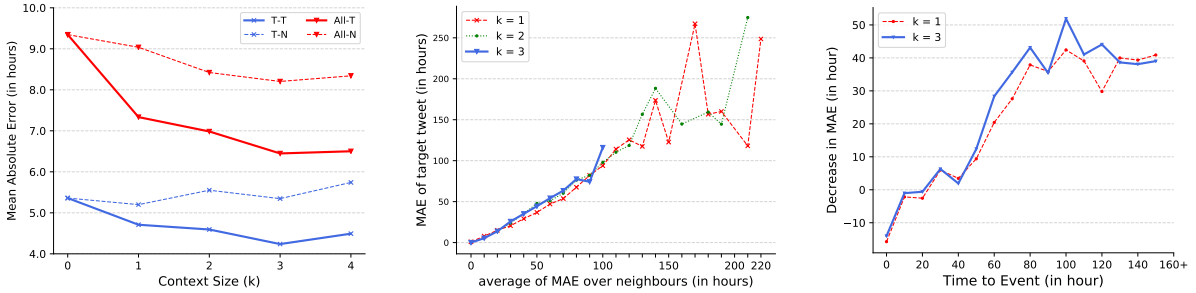
3.2 Settings and Baselines

The hyperparameters of all models are optimized on validation data using random search (Bergstra and Bengio, 2012). We consider **TenseModel** (see below), **Glove**, **BERT**, Event Time Extraction **(ETE)** (Reimers et al., 2018), and **HybridModel** (Hürriyetoğlu et al., 2018) as baselines. TenseModel uses the tense of the outermost verb of a tweet to detect whether it is posted before (+) or after the target event (-). Embedding models are used to represent input tweets and extended to address the time to event task in their last layer. The GLOVE baseline is the model with GLOVE pre-trained embeddings but without context. This baseline has only the attention-weighted average embedding of the target tweet. For BERT baseline, we fine-tuned base version of BERT by adding a linear layer on top for time to event value prediction. ETE uses sentence representation as well as event and position embeddings with a CNN to tackle the target task. They reported a high performance of 84.2% for event status classification on a balanced dataset of news articles. HybridModel is a hybrid of rule-based and data-driven methods focusing on Dutch tweets that carry temporal expressions.

4 Results

4.1 Time to Event Prediction

We compare our context-sensitive model with context size of $k \in \{0, 1, 2, 3\}$ against baseline systems. Mean and Median are heuristic baselines and indicate the mean and median of MAEs of TTE values (i.e., 27.87 and 13.91 respectively). As reported in Table 1, the TenseModel has considerably low performance in distinguishing temporal status of tweets against event times. We attribute this result to the informal language in user-generated content and multi-verb tweets which can challenge the tense model. BERT embeddings slightly improves the performance of other embedding models. However, BERT and ETE's performance are considerably lower than the performance of our model achieved by adding context information ($k \geq 1$). This result indicates that adding neighbouring tweets leads to more accurate prediction of TTE than incorporating better word embeddings. Our model achieves an MAE of 6.43 hours on *All-T* set and 4.24 on *T-T* set (see Table 1). We also compare our model against the Hybrid Model on the



(a) Effect of context size on MAE (b) Accurately-predicted neighbours (c) Farther vs. closer to event instances

Figure 3: (a): Greater context size leads to better estimation of TTE. (b): More accurately-predicted neighbours lead to more accurate estimation of TTE. (c): Context information better help farther to event target instances.

	TTE sign			TTE value
Model	**P**	**R**	**F1**	**MAE** (hours)
Trained and evaluated on the *EPL* dataset				
Mean	-	-	-	27.87
Median	-	-	-	13.91
TenseModel	0.23	0.37	0.28	-
GLOVE (Pennington et al., 2014)	0.73	0.66	0.69	8.43
BERT (Devlin et al., 2019)	0.68	0.79	0.73	7.71
ETE (Reimers et al., 2018)	0.88	0.59	0.70	7.86
Our model ($k = 0$)	0.61	0.52	0.56	9.34
Our model ($k = 1$)	0.73	0.77	0.74	7.31
Our model ($k = 2$)	0.81	**0.87**	0.83	6.98
Our model ($k = 3$)	**0.92**	0.83	**0.87**	**6.43**
Trained on *EPL* and evaluated on the *Oscars* dataset				
BERT	0.46	0.48	0.47	14.2
Our model ($k = 0$)	0.38	0.49	0.43	14.76
Our model ($k = 1$)	0.51	0.57	0.54	13.43
Our model ($k = 2$)	0.55	0.60	0.57	13.37
Our model ($k = 3$)	0.58	0.64	0.61	13.18

Table 1: Model performance in terms of macro precision, recall and F1 for sign classification (TTE sign), and Mean Absolute Error (MAE) for TTE prediction (TTE value) on EPL and Oscars datasets.

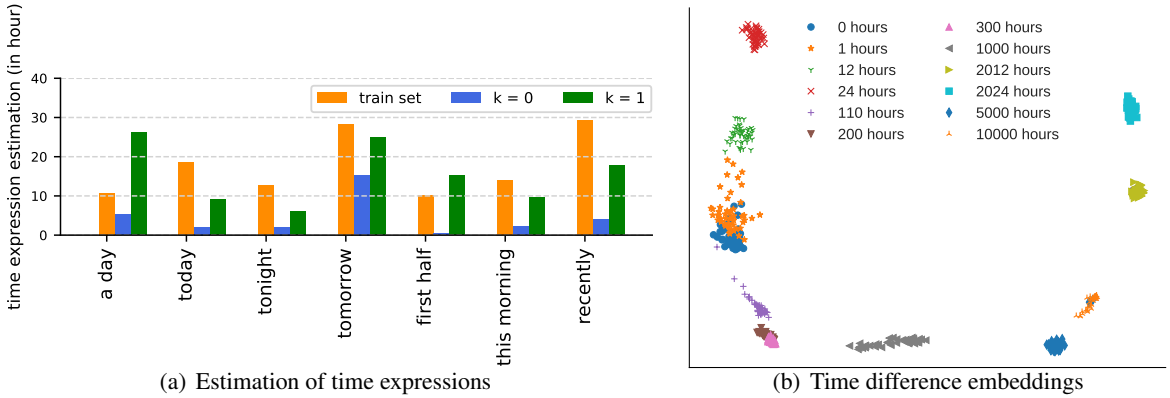
Dutch dataset (see Section 3.1).[6] Using the Dutch embeddings of (Tulkens et al., 2016), our model achieves an MAE of 4.7 hours based on leave-one-out cross validation, while the corresponding value for the Hybrid Model is 8 hours. Evaluation results on Oscars dataset reveals that the model learns how to utilize information of neighbouring tweets and time differences. The lower performance on the Oscar dataset is due to differences in training (EPL) and test (Oscar) data distributions.

Can context information help? To investigate the effect of adding context, we start with a standalone base model that predicts the time to event by just relying on its own content, i.e, $k = 0$. Figure 3(a) illustrates that the performance is higher for tweets that contain at least one time expression (*T set*) compared to *All set*. Accordingly, as we gradually add more context tweets, the performance consistently increases with greater improvement with the *T set* as context. The best preforming model is achieved by adding context of size 3 from *T set*, leading to the lowest time to event estimation error of 4.24 hours.

We also note that context tweets that do not contain any temporal expression (the *N set*) slightly increase the performance; see the dashed lines in Figure 3(a). We conjecture that these tweets add lexical clues that carry implicit temporal information about events. In addition, Figure 3(b) shows a strong correlation between the average error in model prediction performance on context and target tweets. This result shows that neighboring tweets that are more accurately learned by the network are better candidates to use as context for other tweets.

Does context information lead to more accurate estimation of time expressions? To answer this question, we compute the average time to event for each time expression from both training tweets and predictions for test tweets as $H(\text{TIMEX}_i) = \frac{1}{N}\sum_{t_j \in \mathcal{S}, \text{TIMEX}_i \in t_j} \text{TTE}_{t_j}$ where $\mathcal{S} \in \{train, test\}$, TTE_{t_j} indicates time to event for tweet t_j, and N is normalization factor; for training data we use gold values and for test data we use predicted values. Figure 4(a) shows the baseline and estimated values for a range of time expressions. The results show that the value of time expressions are better estimated by adding context. Give that the most frequent time expressions often refer to points in time close to the event (such as *now*) (Hurriyetoglu et al., 2014), our model improves rare time expressions more than the frequent ones, leading to improved prediction of farthest tweets from events.

[6] Note only 71% of 138k tweets are returned by the Twitter API, the rest were deleted or made private by their users.



(a) Estimation of time expressions

(b) Time difference embeddings

Figure 4: (a): Estimation of some selected time expressions. (b): Time differences in the embedding space. Each sample point shows the embedding of time difference between two randomly selected times t_1 and t_2. This figure shows that if $t_1 - t_2 \approx t_1^{'} - t_2^{'}$, their *diff* embeddings are closer in the time difference space.

4.2 Time Difference Embeddings

We generate a synthetic dataset of $2m$ time pairs to evaluate the time difference approach in terms of accurate prediction of time differences between given time pairs, where possible predictions range between 0 to 17520 hours, which corresponds to a maximum difference of two years. Evaluation on $200k$ number of test time pairs shows that the model achieves 98.3% accuracy. In addition, Figure 4(b) shows t-SNE representation of time differences in the embedding space for different randomly selected time pairs. Data points with the same color shows the diff embeddings of the same time differences. The result shows for two random times (t_1, t_2) and $(t_1^{'}, t_2^{'})$, if $t_1 - t_2 \approx t_1^{'} - t_2^{'}$, their *diff* embeddings are very close in the time difference embedding space, indicating the high quality of the resulting space. In addition, Table 2 shows time difference embeddings are useful for TTE estimation since removing them increases the MAE of our full model by a significant amount of 0.8 hours.

4.3 Early prediction

Given that *early* prediction of TTE is more valuable and challenging (due to scarcity of data at earlier times and often imprecise temporal information in earlier tweets), we investigate the performance of our model on target tweets that were posted much earlier than the occurrences of their corresponding events. The results in Figure 3(c) shows that context tweets help farther-to-event instances better than closer ones. This result provides insights for future research on the task of early TTE prediction.

Configuration	MAE (absolute increase)
Full System	6.43
Random diff embeddings	6.64 (+0.21)
No diff embeddings	7.23 (+0.80)
No TTE sign	6.71 (+0.28)

Table 2: Ablation analysis showing changes in Mean Absolute Error (MAE) obtained from removing individual components of the model.

5 Conclusion and Future Work

We developed a context-sensitive neural model that used rich-neighbouring tweets as well as time difference embeddings between target tweets and their neighbors for effective prediction of time to event. We evaluated our and current models on events and tweets of different genres (soccer competitions and academy award ceremonies) and languages (English and Dutch). Future works include expansion to temporal tasks that particularly focus on early prediction of time to events. In addition, it's worth investigating if user or social network information could be helpful for better time to event prediction.

Acknowledgments

We sincerely thank anonymous reviewers for their insightful comments. In addition, this research was completed during the spread of the COVID-19 virus, while the world was in quarantine fearing the epidemic. We would like to dedicate this work to all researchers who contributed to the discovery of the COVID-19 vaccine.

Broader Impact Statement

Our research affects applications that deal with time, and time difference can be an effective feature for them. For example, our work enables automatic creation of calendar of events, which helps keeping individuals informed about potential relevant events. It also help researchers to benchmark their models using our dataset.

In addition, the process of collecting our dataset followed the Twitter policy[7]. We crawled data using the Twitter API and we did not make any attempt to identify any information that have not been volunteered by our user base (e.g., gender, race, wealth, etc.,). We also will just publish the Tweet IDs.

References

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Leo Born, Maximilian Bacher, and Katja Markert. 2020. Dataset reproducibility and ir methods in timeline summarization. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1763–1771.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600k: learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ruihong Huang, Ignacio Cases, Dan Jurafsky, Cleo Condoravdi, and Ellen Riloff. 2016. Distinguishing past, on-going, and future events: The eventstatus corpus. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 44–54.

[7] https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases

Ali Hurriyetoglu, Nelleke Oostdijk, and Antal van den Bosch. 2014. Estimating time to event from tweets using temporal expressions. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 8–16.

Ali Hürriyetoğlu, Nelleke Oostdijk, and Antal van den Bosch. 2018. Estimating time to event of future events based on linguistic cues on twitter. In *Intelligent Natural Language Processing: Trends and Applications*, pages 67–97. Springer.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Douglas R Langbehn, Ryan R Brinkman, Daniel Falush, Jane S Paulsen, MR Hayden, and an International Huntington's Disease Collaborative Group. 2004. A new model for prediction of the age of onset and penetrance for huntington's disease based on cag length. *Clinical genetics*, 65(4):267–277.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the timebank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2195–2204.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2018. Event time extraction with a decision tree of neural classifiers. *Transactions of the Association of Computational Linguistics*, 6:77–89.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eigth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753. ELRA.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3104–3112.

Stephan Tulkens, Chris Emmery, and Walter Daelemans. 2016. Evaluating unsupervised dutch word embeddings as a linguistic resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5310–5318.

Harold I Zeliger. 2016. Predicting disease onset in clinically healthy people. *Interdisciplinary toxicology*, 9(2):39–54.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589.

Improving Compositional Generalization in Classification Tasks via Structure Annotations

Juyong Kim **Pradeep Ravikumar**
Carnegie Mellon University
{juyongk,pradeepr}@cs.cmu.edu

Joshua Ainslie **Santiago Ontañón**
Google Research
{jainslie,santiontanon}@google.com

Abstract

Compositional generalization is the ability to generalize systematically to a new data distribution by combining known components. Although humans seem to have a great ability to generalize compositionally, state-of-the-art neural models struggle to do so. In this work, we study compositional generalization in classification tasks and present two main contributions. First, we study ways to convert a natural language sequence-to-sequence dataset to a classification dataset that also requires compositional generalization. Second, we show that providing structural hints (specifically, providing parse trees and entity links as attention masks for a Transformer model) helps compositional generalization.

1 Introduction

Compositional generalization is the ability of a system to systematically generalize to a new data distribution by combining known components or primitives. For example, assume a system has learned the meaning of "jump" and that "jump twice" means that the action "jump" has to be repeated two times. Upon learning the meaning of the action "jax", it should be able to infer what "jax twice" means. Although modern neural architectures are pushing the state of the art in many complex natural language tasks, these models still struggle with compositional generalization (Hupkes et al., 2020).

In order to advance research in this important direction, in this paper we present two main contributions [1]. First, we present a binary classification dataset which is hard in a compositional way. This allows for studying the compositional generalization ability of a larger range of models than sequence generation tasks, since the task only requires an encoder, and not a decoder. Specifically, we present a methodology to convert an existing semantic parsing dataset, CFQ (Keysers et al., 2019), into a binary classification dataset that is also compositionally hard.

Our second and main contribution is showing that a transformer-based model can better generalize compositionally if we provide hints on the structure of the input. Specifically, we do so by modifying the attention mask used by the model. This is an interesting result, as (except for two additions, which we elaborate on in Section 4) attention masks do not "add" any attention capabilities to the model. Instead, it seems that it is the removal of certain attention pairs that makes the difference. This suggests that vanilla Transformer is having a hard time suppressing non-compositional attention.

2 Background

This section overviews existing work on compositional generalization and then some background on the Transformer models used in this paper. Please see Section B in the appendix for detailed review.

Compositional Generalization. Compositional generalization can manifest in different ways (Hupkes et al., 2020) such as *systematicity* (recombination of known parts and rules) or *productivity* (extrapolation to longer sequences than those seen during training), among others. Early work focused on showing how different deep learning models do not generalize compositionally (Liška et al., 2018), and datasets such as SCAN (Lake and Baroni, 2018) or CFQ (Keysers et al., 2019) were proposed to show these effects.

Work toward improving compositional generalization has proposed ideas such as Syntactic attention (Russin et al., 2019), increased pretraining (Furrer et al., 2020), data augmentation (Andreas, 2019), or general purpose sequential models such as *Neural Turing Machines* or *Differ-*

[1] http://goo.gle/compositional-classification

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 637–645
August 1–6, 2021. ©2021 Association for Computational Linguistics

ential Neural Computers (Graves et al., 2016).

ETC. For our experimental evaluation we use the ETC (Ainslie et al., 2020) Transformer model. ETC extends the standard Transformer model in 3 key ways: (1) it uses a global-local attention mechanism to scale to long inputs, (2) it uses relative attention (Shaw et al., 2018) and flexible masking and (3) it uses a new pre-training loss based on Contrastive Predictive Coding (CPC) (Oord et al., 2018). The last two extensions allow it to handle structured inputs containing, for example, hierarchical structure. In this work, we rely on (2) to annotate the structure of the input.

3 The CFQ Classification Dataset

The Compositional Freebase Questions (CFQ) dataset (Keysers et al., 2019) is an NLU dataset to measure the compositional capability of a learner. It is designed around the task of *translating* a natural language question into a SPARQL query. The dataset has been automatically generated by a grammar and contains 239,357 sentence/query pairs. An example is shown in Figure 3a.

As shown in the original work of Keysers et al. (2019) in order to properly measure the compositional generalization ability of a model, the train and test sets should be split with similar distributions of tokens (*atoms*), but different distributions of their compositions (the *compounds*). In the CFQ dataset, to ensure this, two divergences, namely *atom divergence* and *compound divergence*, between the train and dev/test set are measured while constructing the splits. As a result, carefully selected splits called *maximum compound divergence (MCD)* splits are hard for standard neural networks (they perform well in the train set, but poorly in the test set), while the random splits are easier.

We convert the CFQ dataset into a dataset with a binary classification task. In this new dataset, the input is a question and a SPARQL query, and the task is to determine whether these two sequences have the same meaning or not. Two considerations must be made to ensure the resulting dataset requires compositional generalization:

Negative Example Strategies: Positive instances of the binary classification task can be obtained directly from the original dataset, but to obtain negatives, we use either of two strategies:

- Random negatives: We pair each question with a randomly chosen query.
- Model negatives: Using baseline models (LSTM (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017), and Universal Transformer (Dehghani et al., 2018)) trained on the original CFQ dataset, we get top-k query predictions for each question. After filtering syntactically invalid queries and duplicates, we can get hard examples for classification from their incorrect predictions.

Model negatives are important, as otherwise, the task becomes too easy and would likely not require compositional generalization. See Figure 1 for examples of random/model negative instances.

Compound Distribution of Negative Examples: To prevent data leakage (e.g., compounds from the test set of the original CFQ dataset leaking into the training set of the classification CFQ dataset), we carefully choose the sampling set for random negatives and the train and inference set for model negatives. We generate two splits of the original CFQ dataset. Each split contains three sets with 50% data on train, 25% on dev and 25% on test. The first is a *random split* of the data, and the second (*MCD split*), maximizes the compound divergence between train and dev/test using the same method as in the original CFQ work. Then, we process the examples in each of these sets generating positive and negative examples. For random negatives, we sample negative queries for each questions from the set which the original example belongs to (train/dev/test). For model negatives, to generate negatives for the training set, we divide it into two halves, train models in one, and generate negatives with the other half. For dev/test, we train on dev and generate negatives on test, and vice versa. Figure 2 illustrates this procedure, designed to ensure there is no leakage of compounds between train and dev/test.

For both strategies, we make 1 positive and 3 negatives per original CFQ example. Also, we set aside 5% of the train set as a hold-out set to check i.i.d. generalization.

4 Compositional Generalization via Structure Annotation

Our hypothesis is that part of the difficulty in compositional generalization is to parse the structure of the input. To test this, we evaluate the performance of models when we provide annotations for two structural elements of the inputs: parse

• **Question (CFQ input)**

```
Did M0 ' s writer , editor , director , and cinematographer found M1 and found M2
```

• **Positive query (CFQ output)**

```
SELECT count ( * ) WHERE {
 ?x0 film.cinematographer.film M0 .
 ?x0 film.director.film M0 .
 ?x0 film.editor.film M0 .
 ?x0 film.writer.film M0 .
 ?x0 organizations_founded~ M1 .
 ?x0 organizations_founded~ M2
}
→ Label: 1 (same)
```

• **Model negative query**

```
SELECT count ( * ) WHERE {
 ?x0 film.cinematographer.film M0 .
 ?x0 film.director.film M0 .

 ?x0 film.writer.film M0 .
 ?x0 organizations_founded~ M1 .
 ?x0 organizations_founded~ M2
}
→ Label: 0 (different)
```

• **Random negative query**

```
SELECT DISTINCT ?x0 WHERE {
 ?x0 a film.editor .
 ?x0 film.writer.film M1 .
 ?x0 film.writer.film M2 .
 ?x0 film.writer.film M3 .
 ?x0 ~.person.nationality m_0d060g
}

→ Label: 0 (different)
```

Figure 1: Examples of the CFQ classification dataset. Each query pairs with the question to form an instance. Note the model negative resembles the positive, while the random negative query differs considerably.

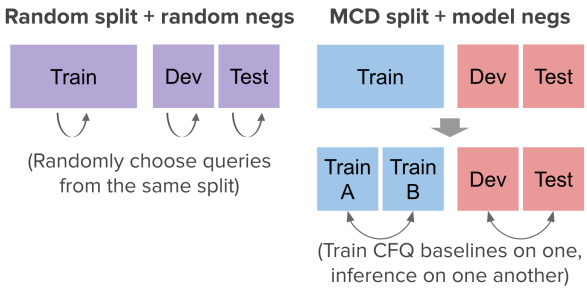


Figure 2: Negative example strategies. Different colors indicate different compound distributions.

trees of both the natural language sentences and SPARQL queries, and *entity cross links* (linking entity mentions from the natural language side to the corresponding mentions in the SPARQL query).

The parse trees of the questions are already given in the original CFQ dataset as constituency-based parse trees. Since the trees include intermediate nodes indicating syntactic structures, we append tokens representing them at the end of each question. We created a simple parser to generate dependency-based parse trees for the SPARQL queries. We join the roots of the two trees to make a single global tree with the `<CLS>` token as the root.

We represent the structure of the inputs by masking attention ("hard mask") or with relative attention (Shaw et al., 2018) labels ("soft mask").

- Hard mask: We customize the binary attention mask of the original Transformer to only allow attention between tokens connected by the edges of the parse tree.
- Soft mask: For every pair of input tokens, we assign relative attention labels based on which of the following edge relationships applies: parent-to-child, child-to-parent, itself, from-or-to-root, or entity-cross-link.

Additionally, we allow attention pairs in the masks connecting the entities appearing both in the question and the queries. We call these links *entity cross links*, and they are found by simple string match (e.g. "`M0`"). Notice that while relative attention labels and the additional tokens to represent the constituency parse tree of the natural language add capabilities to the model, the "hard mask" structure annotations described above (which result in the larger performance gains) do not *add* any attention capabilities to the model. Instead, they simply remove non-structure attention edges. Figure 3b shows the parse trees, and Figure 3c and 3d show the masks for an example.

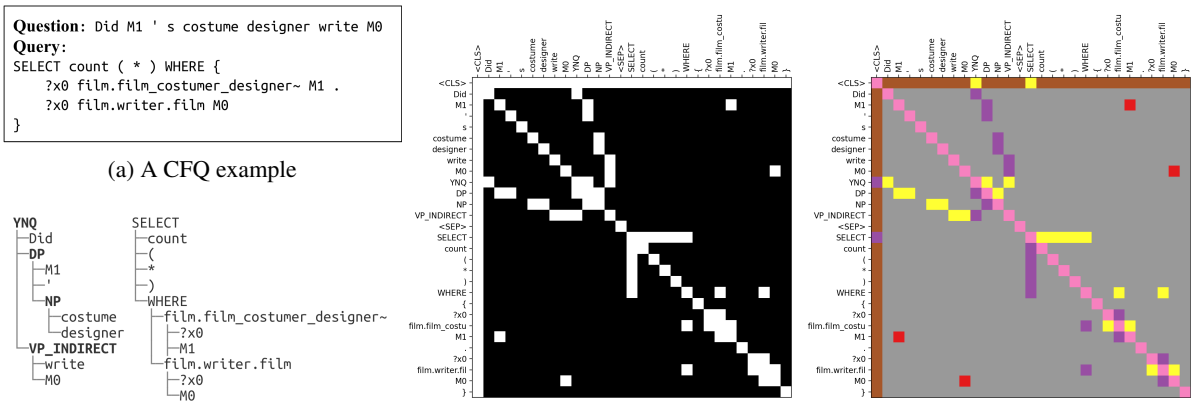
5 Results and Discussion

We used the ETC (Ainslie et al., 2020) Transformer model implementation as it allows us to provide the hard and soft masks described above in an easy way. In all experiments, we report AUC in the dev set as the evaluation metric (we did not evaluate on the test set). Please see Section A in the appendix for training details.

5.1 The CFQ Classification Dataset

We generate two classification datasets: "random split & random negatives" and "MCD split & model negatives", and evaluate LSTM and Transformer models. For both datasets, we evaluate AUC on the hold-out set (taken out of the training set as described above) to test i.i.d. generalization, and on the dev set to test compositional generalization.

As shown in Table 1, models easily generalize on the hold-out set (AUC $\geq$ 0.99). All baseline models also achieve almost 1.0 AUC in the dev set of the "random split & random negatives". However, in the "MCD split & model negatives" models cannot generalize well on the dev set, showing compositional generalization is required. Note that random guessing achieves 0.5 AUC score.



(a) A CFQ example

(b) Parse trees of the CFQ example

(c) Hard mask

(d) Soft mask

Figure 3: Structure annotations for a CFQ example. We extract the hierarchical structure of the question and query of CFQ examples and use them to mask attention (hard mask) and/or provide relative attention labels (soft). Different colors indicate different relative attention labels.

Model	*Random Split & Random Neg*			*MCD Split & Model Neg*		
	Train	**Hold-out**	**Dev**	**Train**	**Hold-out**	**Dev**
LSTM	1.0000	0.9998	0.9998	1.0000	0.9972	0.8310
Transformer (2 layers)	0.9998	0.9997	0.9998	0.9988	0.9931	0.8789
Transformer (6 layers)	0.9999	1.0000	0.9999	0.9995	0.9931	0.8738

Table 1: AUC on the CFQ classification dataset generated with different methods

Model	**Mask Type**	**Cross link**	*MCD Split & Model Neg* **Train**	**Hold-out**	**Dev**
LSTM	-		1.0000	0.9972	0.8310
Transformer	-		0.9995	0.9931	0.8738
Transformer w/ structure annotations (ETC)	No	-	0.9994	0.9934	0.8868
	Hard	N	0.9999	0.9978	0.9061
		Y	1.0000	0.9992	0.9656
	Soft		0.9995	0.9936	0.8819
	Both		1.0000	0.9991	**0.9721**

Table 2: AUC on the CFQ classification dataset (*MCD Split & Model Neg*) with various structure annotations

5.2 Structure Annotation

Table 2 compares different ablations of our structure annotation approach compared to the baseline models. The first (no masks and no cross links) just shows that adding tokens to the input to represent the constituency parsing and moving to ETC only provide small gains (from 0.8738 to 0.8868 AUC). Adding a hard mask already helps the model (0.9061 AUC), and adding cross links on top of that achieves very significant gains (0.9656 AUC). Finally, soft masks by themselves do not seem to help, but a combination of soft and hard masks achieves our best result of 0.9721 AUC.

The interesting result here is that adding the hard mask with entity cross links *only removes* potential attention pairs, so it does not increase model capacity in any way. In other words, the underlying transformer model is in principle able to generalize compositionally to some extent but seems to struggle in suppressing non-compositional attention.

6 Conclusions

The main contribution of this paper is to show that providing structure annotations in the form of attention masks significantly helps Transformer models generalize compositionally. This is interesting for two main reasons: first, it shows that neural network models do have the innate ability to generalize compositionally to some extent, but need some guidance to do so (e.g., by providing attention masks as in our work). This reinforces previous work showing that LSTMs also can, in principle, generalize compositionally, but they just do so with very low probability (Liška et al., 2018). The second reason is that structure annotations, which we provided manually, could be generated by another model in future work. We also presented a procedure for generating classification datasets that require some degree of compositional generalization starting from sequence generation datasets.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.

Jacob Andreas. 2019. Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2018. Universal transformers. *arXiv preprint arXiv:1807.03819*.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Adam Liška, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Jake Russin, Jason Jo, Randall C O'Reilly, and Yoshua Bengio. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

A Full Experimental Results

In this section, we report the full results on the CFQ classification dataset and the structure annotation experiments. In all configurations, multiple evaluation metrics (accuracy, F1 score, and AUC) are computed by averaging the results of two randomly initialized experiments. We test each network using only the val set, not the test set, since the main purpose of the experiment is to compare the compositional generalization ability, not to select best hyper-parameter. Accuracy and F1 score are computed with the threshold 0.5 of the softmax output of label 1.

All the experiments of the CFQ classification datasets were run using the TensorFlow (Abadi et al., 2016) framework. As we explain in the Section 5, we use the ETC Transformer (Ainslie et al., 2020) code for relative position embeddings. For the Transformer implementation, we use the code provided in a Tensorflow tutorial. The training is run on the `n1-highmem-8` instance (52GB RAM, 8 virtual cpus) of Google Cloud Platform, extended with NVIDIA Tesla V100 GPUs.

Hyper-parameters used in the training of neural networks are listed in Table 3. One thing that we want to clarify is that training steps are required number of steps to converge and the training did not last longer than needed. Nevertheless, the experiments with structure annotations required more training steps than LSTM/Transformer, especially when the network is using hard mask. We conjecture that training with the hard mask of parse trees is slow since only a small part of the attention is not masked and hence propagating the gradient via supervision at the `<CLS>` position is slow.

A.1 The CFQ classification Dataset

Table 4 shows the classification results of various methods of generating classification datasets, including one additional configuration (*MCD Split & Random Negatives*). The dataset generated by this new configuration has the train and the dev/test set that have different compound distributions, because it is based on the MCD split. However, because of the method used in generating negative instances (random negatives), the binary classification of correspondence can be easily generalizable to the dev set.

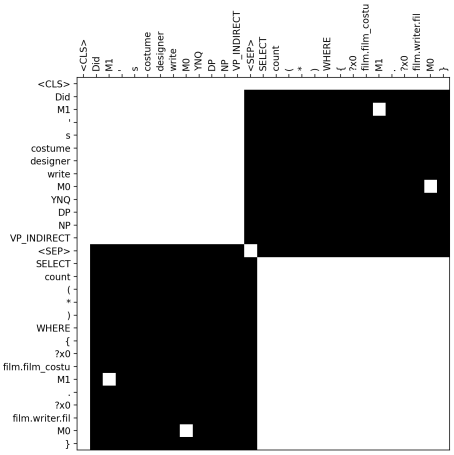


Figure 4: Block attention mask for the CFQ classification example of Figure 3. The dots at top-right and bottom-left are from entity cross links.

A.2 Structure Annotation

One possible annotation of the input structure is a mask to allow tokens of the question and SPARQL queries to only attend within their segment. We call this mask as *block attention* and test it as an alternative to the hierarchical attention structures (parse trees). This mask is denser than the attention mask from parse trees and sparser than "no mask". Figure 4 shows the block attention for the examples shown in the Figure 3.

Table 5 reports the full results of experiments on structure annotations. In all cases, entity cross links improve compositional generalization on the dev set, but provide a significant gain only when combined with the parse tree attention and the attention is guided by the "hard mask". As we can see in the "hard mask" experiments, block attention does not improve compositional generalization, which suggests a need for more detailed attention mask of input structure.

B Related Works on Compositional Generalization

In this section, we review prior works on improving compositional generalization in more detail.

Russin et al. (2019) proposed to split the attention mechanism into two separate parts, syntax and semantics. The semantic part encodes each token independent of the context (this is a pure embedding look-up table), and the syntactic part encodes each token by looking only at its context (without looking at the token itself). In this way, the syntactic part tries to capture the syntactic role a token

	LSTM	Transformer	ETC
Hidden layers	2	{2,6}	6
Last dense layers	2	1	1
Hidden Size	512	128	128
Filter size	-	2048	512
Number of heads	-	16	16
Dropout	0.4	0.1	0.1
Batch size	1024	512	112
Training steps			
Random & Random	20k	10k	-
MCD & Random	20k	10k	-
MCD & Model	30k	20k	200k
Optimizer	Adam (0.85, 0.997)	Adam (0.9, 0.997)	Adam (0.9, 0.997)
Learning rate schedule	Constant	Inverse sqrt	Inverse sqrt
Base learning rate	0.001	0.001	0.001
Warmup steps	-	1000	1000
Weight decay	0.0	0.0	0.0

Table 3: Hyper-parameters used in training deep neural networks on the CFQ classification datasets

might play in a sequence. They show improved compositional generalization on the SCAN dataset using LSTMs, with respect to using standard attention. Compared to Russin et al. (2019) that uses LSTMs for the syntactic part, we use Transformer architecture to handle the hierarchical structure of the input.

In their follow up work on the CFQ dataset, Furrer et al. (2020) showed that an increased amount of pre-training helped Transformer models better generalize compositionally.

Another idea that has been proposed is to augment the training data, adding synthetic training examples to give the model a compositional learning bias (Andreas, 2019) .

Finally, work also exists on using general-purpose models like *Neural Turing Machines* or *Differential Neural Computers* (Graves et al., 2016) that are often trained via reinforcement learning to solve compositional generalization tasks. These models learn an "algorithm" that can solve the task at hand, rather than trying to learn a direct input/output mapping as the Transformer models used in most other works do.

C Examples of the CFQ classification dataset

In Figure 5, we present more examples of the CFQ classification datasets. In all cases, the random negative queries substantially differ from the positive queries, implying that a learner can easily perform the task. On the other hand, the model negative queries only differ by a token or a phrase, which demands a learner's higher ability.

Dataset 1: *Random Split & Random Negatives*

Model	Train			Train (hold-out)			Dev		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
LSTM	0.9999	0.9998	1.0000	0.9984	0.9967	0.9998	0.9982	0.9964	0.9998
Transformer (2 layers)	0.9988	0.9976	0.9998	0.9982	0.9964	0.9997	0.9988	0.9975	0.9998
Transformer (6 layers)	0.9992	0.9988	0.9999	0.9989	0.9978	0.9999	0.9990	0.9979	0.9999

Dataset 2: *MCD Split & Random Negatives*

Model	Train			Train (hold-out)			Dev		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
LSTM	0.9999	0.9998	1.0000	0.9982	0.9965	0.9999	0.9546	0.9025	0.9923
Transformer (2 layers)	0.9982	0.9965	1.0000	0.9974	0.9948	0.9999	0.9942	0.9883	0.9996
Transformer (6 layers)	0.9986	0.9972	0.9999	0.9979	0.9958	0.9997	0.9889	0.9775	0.9991

Dataset 3: *MCD Split & Model Negatives*

Model	Train			Train (hold-out)			Dev		
	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
LSTM	0.9990	0.9979	1.0000	0.9796	0.9604	0.9972	0.8226	0.5199	0.8310
Transformer (2 layers)	0.9817	0.9639	0.9988	0.9592	0.9202	0.9931	0.8359	0.5835	0.8789
Transformer (6 layers)	0.9886	0.9776	0.9995	0.9582	0.9189	0.9931	0.8414	0.6191	0.8738

Table 4: Results of the CFQ classification dataset generated with different CFQ splits and negative example strategies

Model	Mask Type	Parse Tree	Block Attn	Cross link	Train			Train (hold-out)			Dev		
					Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
LSTM	-				0.9990	0.9979	1.0000	0.9796	0.9604	0.9972	0.8226	0.5199	0.8310
Transformer	-				0.9886	0.9776	0.9995	0.9582	0.9189	0.9931	0.8414	0.6191	0.8738
Transformer w/ structure annotations (ETC)	No	-			0.9874	0.9751	0.9994	0.9591	0.9199	0.9934	0.8434	0.6202	0.8868
	Hard	Y	N	N	0.9955	0.9911	0.9999	0.9766	0.9543	0.9978	0.8628	0.6744	0.9061
		Y	N	Y	0.9978	0.9956	1.0000	0.9866	0.9738	0.9992	0.9170	0.8269	0.9656
		N	Y	N	0.9828	0.9659	0.9989	0.9567	0.9152	0.9928	0.8324	0.5874	0.8771
		N	Y	Y	0.9871	0.9746	0.9993	0.9573	0.9171	0.9930	0.8386	0.6048	0.8881
	Soft	Y	N	N	0.9863	0.9728	0.9993	0.9588	0.9197	0.9933	0.8426	0.6017	0.8729
		Y	N	Y	0.9891	0.9784	0.9995	0.9603	0.9226	0.9936	0.8482	0.6385	0.8819
	Hard +Soft	Y	N	N	0.9940	0.9882	0.9999	0.9743	0.9500	0.9973	0.8615	0.6697	0.9056
		Y	N	Y	0.9975	0.9949	1.0000	0.9867	0.9739	0.9991	0.9249	0.8473	**0.9721**

Table 5: Results of the CFQ classification dataset (MCD split & model negatives) with different types of structure annotations

• Question (CFQ input)

```
Did M0 ' s founder produce M1
```

• Positive query (CFQ output)

```
SELECT count ( * ) WHERE {
 ?x0 film.producer.film~ M1 .
 ?x0 ~organizations_founded M0
}
```

→ Label: 1 (same)

• Model negative query

```
SELECT count ( * ) WHERE {
 ?x0 film.producer.film~ M0 .
 ?x0 ~organizations_founded M1
}
```

→ Label: 0 (different)

• Random negative query

```
SELECT count ( * ) WHERE {
 ?x0 a film.film .
 ?x0 film.film.edited_by ?x1 .
 ?x0 film.film.edited_by M3 .
 ?x0 film.film.edited_by M4 .
 ?x0 film.film.written_by M1 .
 ?x0 film.film.written_by M2 .
 ?x1 a film.actor
}
```

→ Label: 0 (different)

(a)

• Question (CFQ input)

```
Did a British sibling of M0 direct M2
```

• Positive query (CFQ output)

```
SELECT count ( * ) WHERE {
 ?x0 film.director.film M2 .
 ?x0 ~.person.nationality m_07ssc .
 ?x0 people.person.sibling_s~ M0 .
 FILTER ( ?x0 != M0 )
}
```

→ Label: 1 (same)

• Model negative query

```
SELECT count ( * ) WHERE {
 ?x0 film.director.film M2 .
 ?x0 ~person.nationality m_07ssc .
 ?x0 people.person.sibling_s~ M2 .
 FILTER ( ?x0 != M0 )
}
```

→ Label: 0 (different)

• Random negative query

```
SELECT count ( * ) WHERE {
 ?x0 a film.editor .
 ?x0 film.cinematographer.film M3 .
 ?x0 ~.person.gender m_05zppz .
 ?x0 ~.person.nationality m_03_3d
}
```

→ Label: 0 (different)

(b)

• Question (CFQ input)

```
Which male person directed M2 , M3 , and M4
```

• Positive query (CFQ output)

```
SELECT DISTINCT ?x0 WHERE {
 ?x0 a people.person .
 ?x0 film.director.film M2 .
 ?x0 film.director.film M3 .
 ?x0 film.director.film M4 .
 ?x0 people.person.gender m_05zppz
}
```

→ Label: 1 (same)

• Model negative query

```
SELECT DISTINCT ?x0 WHERE {
 ?x0 a people.person .
 ?x0 film.director.film M2 .
 ?x0 film.director.film M3 .
 ?x0 film.director.film M4 .
 ?x0 people.person.gender m_02zsn
}
```

→ Label: 0 (different)

• Random negative query

```
SELECT count ( * ) WHERE {
 ?x0 a film.actor .
 ?x0 film.editor.film M1 .
 ?x0 film.editor.film M2 .
 ?x0 ~films_executive_produced M3 .
 ?x0 film.writer.film ?x1 .
 ?x1 a film.film
}
```

→ Label: 0 (different)

(c)

• Question (CFQ input)

```
Who directed a film , executive produced M1 and M2 , and executive produced M3 and M4
```

• Positive query (CFQ output)

```
SELECT DISTINCT ?x0 WHERE {
 ?x0 a people.person .
 ?x0 film.director.film ?x1 .
 ?x0 ~films_executive_produced M1 .
 ?x0 ~films_executive_produced M2 .
 ?x0 ~films_executive_produced M3 .
 ?x0 ~films_executive_produced M4 .
 ?x1 a film.film
}
```

→ Label: 1 (same)

• Model negative query

```
SELECT DISTINCT ?x0 WHERE {
 ?x0 a people.person .
 ?x0 ~films_executive_produced ?x1 .
 ?x0 ~films_executive_produced M1 .
 ?x0 ~films_executive_produced M2 .
 ?x0 ~films_executive_produced M3 .
 ?x0 ~films_executive_produced M4 .
 ?x1 a film.film
}
```

→ Label: 0 (different)

• Random negative query

```
SELECT count ( * ) WHERE {
 ?x0 a film.cinematographer .
 ?x0 film.editor.film M1 .
 ?x0 film.editor.film M2 .
 ?x0 film.editor.film M3 .
 ?x0 ~films_executive_produced M1 .
 ?x0 ~films_executive_produced M2 .
 ?x0 ~films_executive_produced M3 .
 ?x0 film.writer.film M1 .
 ?x0 film.writer.film M2 .
 ?x0 film.writer.film M3
}
```

→ Label: 0 (different)

(d)

Figure 5: Examples of the CFQ classification dataset. Each query pairs with the question to form an instance. Note the model negative resembles the positive, while the random negative query differs considerably. In the model negative queries, the differences from the positive query are marked in bold.

Learning to Generate Task-Specific Adapters from Task Description

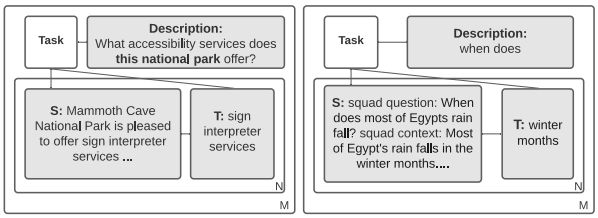
Qinyuan Ye **Xiang Ren**
University of Southern California
{qinyuany,xiangren}@usc.edu

Abstract

Pre-trained text-to-text transformers such as BART have achieved impressive performance across a range of NLP tasks. Recent study further shows that they can learn to generalize to novel tasks, by including task descriptions as part of the source sequence and training the model with (source, target) examples. At test time, these fine-tuned models can make inferences on new tasks using the new task descriptions as part of the input. However, this approach has potential limitations, as the model learns to solve individual (source, target) examples (*i.e.*, at the *instance* level), instead of learning to solve tasks by taking all examples within a task as a whole (*i.e.*, at the *task* level). To this end, we introduce HYPTER, a framework that improves text-to-text transformer's generalization ability to unseen tasks by training a hypernetwork to generate task-specific, light-weight adapters from task descriptions. Experiments on ZEST dataset and a synthetic SQuAD dataset demonstrate that HYPTER improves upon fine-tuning baselines. Notably, when using BART-Large as the main network, HYPTER brings 11.3% comparative improvement on ZEST dataset.[1]

1 Introduction

Pre-trained text-to-text models (Raffel et al., 2020; Lewis et al., 2020) provide a unified formulation and off-the-shelf weights for a variety of NLP tasks, such as question answering (Khashabi et al., 2020) and commonsense reasoning (Bosselut et al., 2019). In addition to their strong performance, text-to-text models naturally support generalizing to novel tasks, by incorporating task description as part of the source sequence and fine-tuning the model with (source, target) examples (Weller et al., 2020). At inference time, the model is required to perform unseen tasks with the source sequence containing new task descriptions.



(a) Zero-shot Learning from Task Description, ZEST dataset (Weller et al., 2020)

(b) Synthetic Version of SQuAD, (Rajpurkar et al., 2016)

Figure 1: Instead of learning from (source, target) examples, in this paper we study the problem of *learning from task descriptions* (Weller et al., 2020). The train set contains M tasks, and the i-th task contains N_i examples of (s, t) pairs in text format. During test time, the learned model is required to directly make inferences on a new task given a task description.

While this initial attempt shows positive results, there are two potential limitations for the direct fine-tuning approach. (1) Predictions can be sensitive to the task descriptions (or "prompts") that are heuristically designed (Jiang et al., 2020). Paraphrasing the task description may lead to performance downgrade. (2) The model still learns from individual (source, target) examples, instead of learning to solve tasks at a higher level, by explicitly taking multiple examples within a task as a whole (see Fig. 1). Meanwhile, applying existing zero-shot learning methods that supports task-level learning to text-to-text transformers is non-trivial. Methods designed specifically for classification problems, such as prototypical networks (Snell et al., 2017), cannot be directly applied to text-to-text models. Moreover, given the large size of text-to-text models, generating parameters for a whole model from the task description (Jin et al., 2020) is infeasible.

In this work, we follow the settings in (Weller et al., 2020) and aim to improve a model's generalization ability to unseen tasks by better incorporating task descriptions and using a task-level training procedure. We introduce HYPTER, a frame-

[1]Code and data can be found at https://github.com/INK-USC/hypter.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 646–653
August 1–6, 2021. ©2021 Association for Computational Linguistics

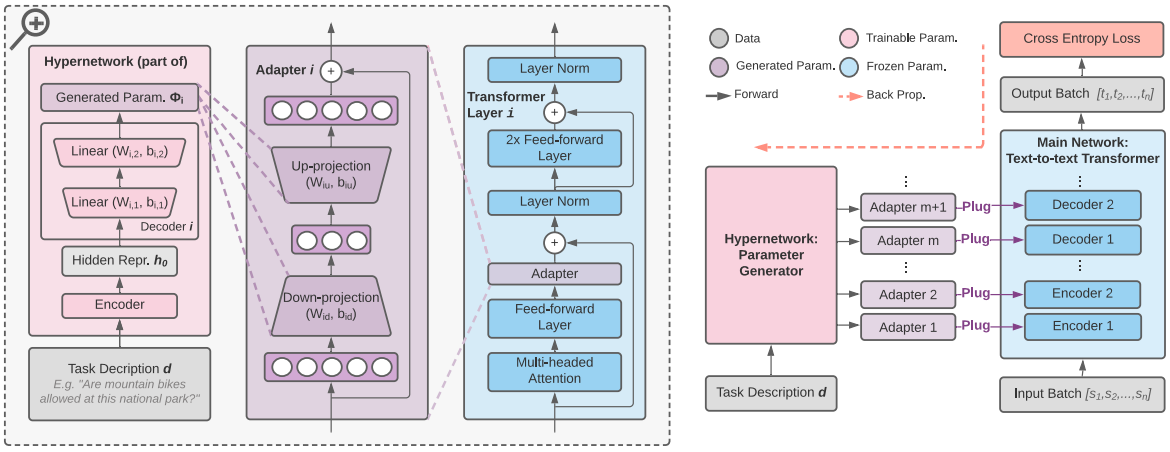


Figure 2: **Illustration of HYPTER Framework.** **Left:** A hypernetwork generates parameter ϕ_i for task-specific adapter i that is plugged to transformer layer i in the text-to-text model. **Right:** The adapted main network is evaluated on a task $(d, \mathcal{D})$. The final cross entropy loss is back-propagated to update the hypernetwork.

work that employs a hypernetwork (Ha et al., 2017) to dynamically generate task-specific parameters (*i.e.*, adapters) from task descriptions. Adapters (Houlsby et al., 2019) are light-weight modules that can be inserted into transformer layers for *parameter-efficient* adaptation. Such formulation also effectively enables learning at the *task* level, by learning to generate appropriate parameters for a task, and examine the model's competence on each task using multiple examples within that task. This is in contrast to learning at the *instance* level, by learning to generate the correct output for one specific input sequence.

We apply HYPTER to two datasets: ZEST (Weller et al., 2020) and a synthetic version of SQuAD (Rajpurkar et al., 2016). We demonstrate that HYPTER improves upon direct fine-tuning baselines. Notably, training with HYPTER achieves 0.45% absolute improvement (11.3% comparative improvement) in Competence@90 metric on ZEST, when BART-Large is used as the main network.

2 Problem Definition

We study the problem of *learning from task description* (Weller et al., 2020), and aim to improve models' competence on unseen tasks at the inference time. Formally, a task is denoted as a tuple of $(d, \mathcal{D})$, where d is the natural language description of the task, and $\mathcal{D} = \{(s_1, t_1), ..., (s_n, t_n)\}$ contains (source, target) examples of this task (See Fig. 1). In our text-to-text formulation, both s_i and t_i are text sequences. At train time, both d and $\mathcal{D}$ are available, while at test time, an unseen description d is given, and the model is expected to predict the correct t given input s without further training.

For instance, in the ZEST dataset (Weller et al., 2020), a train task description can be "*Are mountain bikes allowed at this national park?*", while $\mathcal{D}$ contains twenty paragraphs for different national parks and twenty corresponding answers. During test time, a novel task may be "*Are there fish in this national park that live in caves?*", and the model is asked to directly make inferences.

3 Background: Adapters

Our work is built on adapters (Houlsby et al., 2019), light-weight modules that can be placed into transformer layers for parameter-efficient transfer learning. In the original paper, the main model is frozen during training, while only layer norm and adapter parameters are learnable. In this paper, we adopt a simplified design compared to the original paper (see Fig. 2 (Left)) – In each transformer layer, exactly one adapter module will be added after the multi-headed attention. One adapter module contains two linear layers separated by an non-linearity activation layer. We use $(\mathbf{W}_{id}, \mathbf{b}_{id})$ to denote the down-projection parameters for the adapter in transformer layer i, and $(\mathbf{W}_{iu}, \mathbf{b}_{iu})$ for the up-projection parameters.

4 Method

Overview. Fig. 2 provides an illustration of our HYPTER framework. HYPTER has two major parts: (1) A main network, which is a pre-trained text-to-text model. We instantiate the main network with BART-Base/Large (Lewis et al., 2020). (2) A hyper-

network, which generates adapters to be plugged into the main network. Fig. 2 (Left) contains a detailed illustration of how adapter parameters are generated and how adapter layers are incorporated into one transformer layer.

Hypernetwork. The hypernetwork consists of an encoder and multiple decoders. The encoder maps the task description d to a latent representation $\mathbf{h}_0$, while the decoders use $\mathbf{h}_0$ to generate adapter parameters ϕ. In our work we instantiated the encoder with a RoBERTa-Base model (Liu et al., 2019), *i.e.*, $\mathbf{h}_0 = \text{RoBERTa}(d)$. For a text-to-text model with n transformer layers, the hypernetwork contains n decoders. Decoder i uses $\mathbf{h}_0$ as input, and outputs adapter parameters ϕ_i for transformer layer i, *i.e.*, $\mathbf{h}_{i,1} = \text{ReLU}(\mathbf{W}_{i,1}\mathbf{h_0} + \mathbf{b}_{i,1})$, $\phi_i = \mathbf{W}_{i,2}\mathbf{h}_{i,1} + \mathbf{b}_{i,2}$. Here $\mathbf{W}_{i,1}, \mathbf{b}_{i,1}, \mathbf{W}_{i,2}, \mathbf{b}_{i,2}$ are trainable parameters. The generated parameters ϕ_i are sliced and reshaped to become parameters $[\mathbf{W}_{id}, \mathbf{b}_{id}, \mathbf{W}_{iu}, \mathbf{b}_{iu}]$ used in the adapter i.

Model Training. We adopt a training schedule where we first train the main network, then train the hypernetwork while the main network is frozen. Conceptually, the first stage ensures that the main network captures the general ability across different tasks; the second stage allows the hypernetwork to learn to adapt the main network to a specific task. During the first stage the text-to-text model is fine-tuned with all $(\text{Concat}(d, s), t)$ examples in the training set. Here $\text{Concat}(d, s)$ means the concatenation of task description d and input s. The learned main network from this stage also serves as the baseline method.

During the second stage, we sample a task $(d, \mathcal{D})$ from the training set and sample a mini-batch of (s, t) examples from $\mathcal{D}$. Given a description d, the hypernetwork generates adapter parameters ϕ_i. We insert the resulting adapter layers into the main network, and compute the cross entropy loss L of generating t given input $\text{Concat}(d, s)$. The loss is end-to-end differentiable and is back-propagated to update the hypernetwork, while the main network is frozen. See Fig. 2 (Right) for illustration. This second stage of training effectively enables learning at the *task* level. The loss L characterizes the model's competence in the task $(d, \mathcal{D})$. Therefore, by optimizing L, the model is trained to *solve tasks*.

Model Inference. At test time the model is given an unseen task description d. The hypernetwork generates description-dependent adapter parameters, similar to the procedure during training. In this way, we obtain a model that is capable of making inferences for this new task.

5 Experiments

5.1 Experiment Setup

Datasets. We use two datasets that fit our setup. The first one is Zero-shot Learning from Task Descriptions dataset (ZEST, Weller et al. 2020), which formulates task descriptions as generalized questions, and provides multiple source-target examples for each question. The performance is evaluated with a novel metric: "Competence@K", along with mean F1 score. Competence@K is the percentage of all tasks for which the model achieves mean F1 score higher than K. For example, Competence@90=5 suggests that 5% of all tasks can be solved with mean F1 better than 90%. We report dev set performance, and hidden test set performance obtained from ZEST's official leaderboard.

We construct the second dataset from SQuAD v1 (Rajpurkar et al., 2016) to simulate the problem setting in this paper. We refer to this dataset as Synthetic SQuAD. Specifically, we construct tasks from the original SQuAD train set according to "question type", the bi-gram containing the central question word (*e.g.*, what, when). For example, "when does" questions are considered as a task, and "what country" questions are considered as another task. These bi-grams are used as "task descriptions". We select the 100 most frequent question types in SQuAD train set, and randomly subsample 64 examples from each type to formulate our dataset. We then randomly split the 100 types into 80/10/10 for train/dev/test. In addition, we select examples that fall into the 10 test question types from Natural Questions (Kwiatkowski et al., 2019) and NewsQA (Trischler et al., 2017), and use these as out-of-domain test examples. Performance is evaluated with mean F1. We include the list of question types and more details about this dataset in Appendix A.

Baseline. To demonstrate the efficacy of the HYPTER framework, we compare it to just its first half – the main text-to-text transformer model that we obtain after the first stage of training. This is identical to the fine-tuning baseline method in (Weller et al., 2020), and there are no other applicable baselines to the best of our knowledge.

Model	Mean-F1	C@75	C@90
Bart-Base	28.44 (±1.58)	5.76 (±2.10)	0.74 (±0.00)
+ HYPTER	**28.96** (±1.15)	**6.32** (±2.02)*	**1.08** (±0.62)
Bart-Large (reported)	40	13	8
Bart-Large	41.17 (±1.16)	15.74 (±2.16)	7.17 (±1.66)
+ HYPTER	**41.65** (±1.34)	**16.41** (±2.15)*	**7.62** (±1.66)*

Table 1: **Performance on ZEST Dev Set.** "C@75/90" refers to Competence@75/90 metric. We report mean and standard deviation over 7 runs. $*$ indicates statistical significance in a two-tailed paired t-test ($p < 0.05$).

Model	Mean-F1	C@75	C@90
Bart-Base	31.97	**7.03**	2.23
+ HYPTER	**32.32**	6.72	**2.53**
Bart-Large (reported)	37.93	11.19	3.96
Bart-Large	40.13	10.91	3.98
+ HYPTER	**40.41**	**11.35**	**4.43**

Table 2: **Performance on ZEST Test Set.** Performance obtained from ZEST official leaderboard[2].

Training Details. For each method, we train the model 7 times using different random seeds, and we report average and standard deviation. We discuss other training details, including hyperparameters, in Appendix B. Notably, we ensure all baseline models will not benefit from additional training, by tuning the number of epochs and using early stopping based on dev performance. This ensures the improvement brought by HYPTER is not due to additional training.

5.2 Results

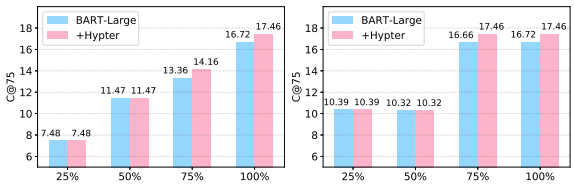
Main Results. We present the results for ZEST in Table 1-2 and results for Synthetic SQuAD in Table 3. On ZEST test set, we observe that the Competence@90 metric is improved from 3.98 to 4.43 when using BART-Large, yielding an 11.3% relative improvement. When BART-Base is used, C@90 is improved from 2.23 to 2.53. This demonstrates that by learning to solve tasks with HYPTER, the model's generalization ability to unseen tasks is improved. On Synthetic SQuAD dataset, we observe 0.74% improvement with BART-Base and 0.41% improvement with BART-Large. Additionally, models trained with HYPTER achieves comparable or better performance on out-of-domain test sets, suggesting the learned task-solving ability is generalizable to new test distribution.[3] It is a known issue that evaluating zero-shot performance can be tricky. We tried our best to reduce the randomness and instability by using different random seeds. In Table 1 and Table 3, we demonstrate that performance improvement is significant (p<0.05) in multiple settings, *e.g.*, on ZEST dev set when C@75 metric is used.

[2] https://leaderboard.allenai.org/zest/submissions/public

[3] Unexpectedly, in Table 3 we observe that performance of BART-Large on NewsQA is worse than that of BART-Base. We suspect that BART-Large may have overfit the SQuAD train set during the first stage of fine-tuning.

Model	SQuAD	NQ	NewsQA
Bart-Base	74.79 (±0.91)	49.78 (±0.95)	56.37 (±0.90)
+ HYPTER	**75.53** (±0.68)*	**50.39** (±1.01)*	**56.41** (±0.85)
Bart-Large	79.32 (±0.34)	59.21 (±0.89)	55.41 (±0.54)
+ HYPTER	**79.73** (±0.50)	**59.58** (±0.57)	**55.60** (±0.90)

Table 3: **Performance on Synthetic SQuAD dataset.** We report mean and standard deviation over 7 runs. NQ and NewsQA serve as out-of-domain test data.



(a) # Tasks (b) # Examples per Task

Figure 3: Competence@75 Performance on ZEST Dev when less training data is used.

Model Behavior Analysis on ZEST. ZEST dataset provides a comprehensive analysis protocol by splitting tasks into different generalization types (base, paraphrase, composition, semantic flips, and output structure) and defining four error types (recall, precision, partial, and other). Compared to the BART-Large fine-tuning baseline, our model achieves better performance in "base" and "paraphrase" categories in the ZEST official test set. We also manually inspected dev set predictions produced by the baseline and our model. We found the predictions corrected by our method span across the four error types. In particular, the proposed method flipped two "n/a" predictions into the correct answers in the task "Which royalty was this dog breed popular with?" ("base" category), reducing the recall errors and improving the competence metric. We do not observe more granular model behavioral patterns beyond this point.

Study of Data Efficiency. We study whether HYPTER is effective when trained with (1) fewer tasks, while the number of examples per task is unchanged; (2) fewer examples per task, while the number of total tasks is kept constant. We experiment with ZEST and BART-Large, and show the performance in Fig. 3. We observe that HYPTER is effective when trained with 75%/100% tasks, but does not improve performance with fewer tasks. This is reasonable since HYPTER learns at the *task*

level (taking one task as an "example"), and 50% of the tasks may be insufficient. We also observe performance improvement with 75%/100% examples per task, but not with fewer examples. This suggests sufficient number of examples per task is necessary for HYPTER to generate effective adapters.

6 Related Work

Zero-shot Learning with Transformers. Zero-shot learning (ZSL) has been explored for various NLP tasks, including text classification (Yin et al., 2019), entity linking (Logeswaran et al., 2019) and entity typing (Obeidat et al., 2019). Several works study cross-task transfer by unifying the input-output format, *e.g.*, relation extraction as machine reading comprehension (Levy et al., 2017), named entity recognition as machine reading comprehension (Li et al., 2020). Such formulation allows generalization to unseen relation or named entity types at test time. Learning from task descriptions (Weller et al., 2020) and instructions (Mishra et al., 2021) can be considered as a sub-category in zero-shot learning, with the goal of generalizing to unseen tasks during inference.

Adapters for Transformers. Houlsby et al. (2019) proposed adapter layers for parameter-efficient transfer learning in NLP. Adapter layers, which adopt a bottleneck architecture with two linear layers, are added after each multi-headed attention layer and each feed-foward layer in a pre-trained transformer. Adapters have been recently applied to multi-lingual settings, with successes in NER, QA and commonsense reasoning (Pfeiffer et al., 2020; Philip et al., 2020; Artetxe et al., 2020).

Hypernetworks and Contextual Parameter Generators. Hypernetwork (Ha et al., 2017) is a broad concept of "using one network to generate the weights for another network". This concept has been broadly applied to visual reasoning (Perez et al., 2018), zero-shot image classification (Jin et al., 2020), etc. Closely related to our work, UDapter (Üstün et al., 2020) studies multilingual dependency parsing by generating adapter parameters. Our work is more generalizable as we do not restrict task format (dependency parsing v.s. general text-to-text tasks) or relations between subtasks (cross-lingual tasks v.s. tasks with text-form descriptions).

7 Conclusion

In this paper, we introduced HYPTER, a framework to improve text-to-text transformer's generalization ability to unseen tasks. HYPTER enhances task-specific abilities by inserting adapters generated with a hypernetwork, meanwhile it maintains the model's general task-solving ability by freezing main model parameters. We demonstrated the effectiveness of HYPTER on two datasets. Future work may explore teaching models with compositional instructions using HYPTER, or propose robust fine-tuning methods that help the model generalize to unseen data. It is also necessary to construct a large dataset of diverse NLP tasks to facilitate future research in this direction.

Acknowledgments

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007, the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, the Defense Advanced Research Projects Agency with award W911NF-19-20271, and NSF SMA 18-29268. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. We would like to thank anonymous reviewers and collaborators in USC INK research lab for their constructive feedback.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019

shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

David Ha, Andrew M. Dai, and Quoc V. Le. 2017. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Tian Jin, Zhun Liu, Shengjia Yan, Alexandre Eichenberger, and Louis-Philippe Morency. 2020. Language to network: Conditional parameter adaptation with natural language descriptions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6994–7007, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hanna Hajishirzi. 2021. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *ArXiv*, abs/2104.08773.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Description-based zero-shot fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 807–814, Minneapolis, Minnesota. Association for Computational Linguistics.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3942–3951. AAAI Press.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages

4465–4470, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

A Dataset Details

ZEST. ZEST dataset is released at https://ai2-datasets.s3-us-west-2.amazonaws.com/zest/zest.zip. ZEST leaderboard is hosted at https://leaderboard.allenai.org/zest/submissions/public.

Synthetic SQuAD. We build our synthetic dataset from the processed version of SQuAD, Natural Questions and NewsQA in MRQA Shared Task 2019 (Fisch et al., 2019) (https://mrqa.github.io/2019/). We provide the script to reconstruct the data we use in our released code. We list the bigrams we use to formulate synthetic tasks and their train/dev/test partition in Listing 1.

Listing 1: Train/Dev/Test Partition in Synthetic SQuAD dataset.

```
"train": ["why were", "what years", "who said", "what percent", "when did", "where do", "who is", "how are", "what decade", "how does", "how long", "where was", "what has", "which two", "who was", "who were", "where are", "where does", "what did", "how far", "what organization", "what does", "what group", "what would", "how did", "who has", "who created", "how many", "what name", "what types", "what two", "which city", "who are", "how is", "what event", "what are", "what century", "what area", "whom did", "why was", "who wrote", "why are", "where is", "how old", "when is", "what caused", "who did", "where did", "what happened", "what state", "what kind", "what time", "what famous", "what's the", "what day", "what is", "what company", "what were", "why do", "what new", "what date", "what do", "what color", "which group", "what country", "how can", "what have", "where can", "what period", "which year", "when was", "what other", "what happens", "was the", "what was", "which of", "when were", "what sort", "what city", "what year"],
"dev": ["what month", "why is", "what part", "what term", "how was", "how were", "how do", "who led", "which country", "when does"],
"test": ["where were", "what political", "what religion", "why did", "what type", "what language", "who had", "what percentage", "what can", "how much"]
```

B Training Details

We use transformers (Wolf et al., 2020) for all our experiments. All experiments are done with one single GPU. We use NVIDIA Quadro RTX 8000, NVIDIA Quadro RTX 6000, or NVIDIA GeForce RTX 2080 Ti, depending on availability.

For text-to-text model fine-tuning, we select learning rate from {1e-5, 3e-5, 5e-5}, and select the total number of epochs from {5, 10, 15, 20, 30} for ZEST and {10, 20, 30, 50, 100} for synthetic SQuAD. We use a fixed batch size of 32.

For hypernetwork training, we train up to 100 epochs (one epoch here refers to an iteration over all tasks). We update the hypernetwork every b tasks, and we call b as task batch size. When learning from one task, we sample b' examples within this task, and we call b' as the example batch size. We greedily and sequentially select adapter width d from {4,8,16,32}, learning rate α from {3e-6, 1e-5, 3e-5, 1e-4}, b from {4,8,16,32}, b' from {4,8,16,32}, based on dev set performance.

C Additional Baseline

Another reasonable baseline is to fine-tune a text-to-text model together with randomly initialized adapters plugged in it. We experiment with this method using BART-Large and list the performance in Table 4. We do not observe significant differences between the two methods (p=0.8840 for C@75, p=0.8118 for C@90 in two-tailed paired t-test).

Model	Mean-F1	C@75	C@90
Bart-Large	41.17 (±1.16)	15.74 (±2.16)	7.17 (±1.66)
Bart-Large with Adapters	39.76 (±1.26)	15.61 (±1.14)	6.96 (±1.15)

Table 4: Performance comparison when adapters are plugged / not plugged during fine-tuning.

D Dev Set Performance of Models Submitted to ZEST Leaderboard

In Table 5 we present the dev performance of models submitted to the leaderboard. The submitted models are the "first-runs" in the 7-run series, as we add the 7-run experiments and significance test later on, following a reviewer's suggestion.

Model	Mean-F1	C@75	C@90
Bart-Base	29.72	7.87	**4.05**
+ HYPTER	**29.81**	**8.67**	**4.05**
Bart-Large (reported)	40	13	8
Bart-Large	42.10	16.72	8.85
+ HYPTER	**43.50**	**17.46**	**9.64**

Table 5: Dev set performance of models submitted to ZEST leaderboard.

E Discussion

It is worth noting that the efficacy of HYPTER is at the cost of introducing new parameters in the hypernetwork. To generate adapter parameters, more parameters are introduced and trained in the hypernetwork. One may achieve better generalization ability to unseen tasks with larger pre-trained models with billions of parameters. In this case, we consider HYPTER as an alternative by augmenting a medium-sized pre-trained model with a hypernetwork. Meanwhile, we highlight our contribution to be the concept of generating task-specific adapters from descriptions and HYPTER's task-level training procedure.

QA-Driven Zero-shot Slot Filling with Weak Supervision Pretraining

Xinya Du*
Cornell University
xdu@cs.cornell.edu

Luheng He
Google Research
luheng@google.com

Qi Li
Google Assistant
qilqil@google.com

Dian Yu*
University of California, Davis
dianyu@ucdavis.edu

Panupong Pasupat
Google Research
ppasupat@google.com

Yuan Zhang
Google Research
zhangyua@google.com

Abstract

Slot-filling is an essential component for building task-oriented dialog systems. In this work, we focus on the zero-shot slot-filling problem, where the model needs to predict slots and their values, given utterances from new domains without training on the target domain. Prior methods directly encode slot descriptions to generalize to unseen slot types. However, raw slot descriptions are often ambiguous and do not encode enough semantic information, limiting the models' zero-shot capability. To address this problem, we introduce QA-driven slot filling (QASF), which extracts slot-filler spans from utterances with a span-based QA model. We use a linguistically motivated questioning strategy to turn descriptions into questions, allowing the model to generalize to unseen slot types. Moreover, our QASF model can benefit from weak supervision signals from QA pairs synthetically generated from *unlabeled* conversations. Our full system substantially outperforms baselines by over 5% on the SNIPS benchmark.

1 Introduction

Automatic slot filling, which extracts task-specific slot fillers (e.g. flight date, cuisine) from user utterances, is an essential component to spoken language understanding (Bapna et al., 2017). As shown in Figure 1, the model predicts the slot filler "Joe A. Pass" for the slot type "artist" given an input utterance. However, fully supervised slot filling models (Young, 2002; Goo et al., 2018) require labeled training data for each type of slot (Shah et al., 2019). It is even more of a problem for data-intensive models (Mesnil et al., 2014). This makes the development of new domains in these systems a challenging and resource-intensive task.

This has motivated studies in cross-domain zero-shot learning for the slot-filling task (ZSSF), where the goal is to achieve good slot-filling performance on new domains without requiring additional training data. Previous work (Bapna et al. (2017); Shah et al. (2019)) often uses a sequence tagging approach (similar to the upper image in Figure 1 in a high-level way). To achieve zero-shot domain transfer, they directly encode raw *slot descriptions or names*, such as "playlist", "music item", to enable models to generalize to slot types unseen at training time. However, slot descriptions are often ambiguous and typically do not encode enough semantic information by themselves.

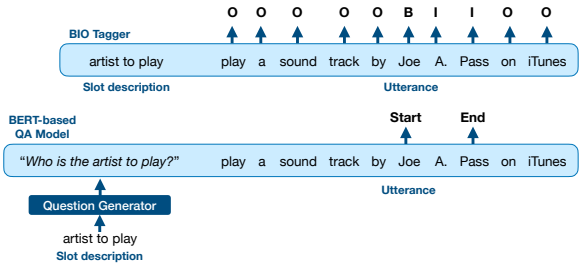


Figure 1: Comparison between two slot-filling frameworks: BIO tagging based model (upper) and our QA-based model (below).

Instead of directly encoding slot descriptions and examples, we introduce a QA-driven slot filling framework (QASF) (Figure 1). Inspired by the recent success of QA-driven approaches (McCann et al., 2018; Logeswaran et al., 2019; Gao et al., 2019; Li et al., 2020; Namazifar et al., 2020), we tackle the slot-filling problem as a reading comprehension task, where each slot type (e.g. "artist") is associated with a natural language question (e.g. "Who is the artist to play?"). A span-based reading comprehension model is then used to extract a slot filler span from the utterance by answering the question.[1] In this work, we use a linguisti-

*Work done during internship at Google Research.

[1]It can been seen as an extension of the QA-driven meaning representations (He et al., 2015; Michael et al., 2017), where

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 654–664
August 1–6, 2021. ©2021 Association for Computational Linguistics

cally motivated question generation strategy for converting slot descriptions and example values into natural questions, followed by a BERT-based QA model for extracting slot fillers by answering questions. As shown in our experiments, this QA-driven method is better at exploiting the semantic information encoded in the questions, therefore it generalizes better to new domains without any additional fine-tuning, as long as the questions are meaningful enough. To the best of our knowledge, we are the first to leverage weakly supervised synthetic QA pairs extracted from unlabeled conversations for a second-stage pretraining. Drawing insights from Mintz et al. (2009), we create a weakly supervised QA dataset from unlabeled conversations and an associated ontology. The synthetic QA pairs are constructed by matching unlabeled utterances against possible slot values in the ontology. This provides a general and cost-effective way to improve QA-based slot filling performance with easily obtainable data.

Experimental results show that (1) our QASF model significantly outperforms previous zero-shot systems on SNIPS (Coucke et al., 2018) and TOP (Gupta et al., 2018); (2) encoding natural questions help models better leverage weakly supervised signals in the pretraining phase, compared to encoding raw descriptions.

2 Task Definition

Given an input utterance u, a slot filling model extracts a set of (slot type, span) pairs $(s_i, a_i), i = 1, \ldots, k$ where s_i comes from a fixed set of slot types $\mathcal{S}$, and each $a_i = (j, k), 1 \leq j < k \leq |u|$ is a span in u. Each slot type is accompanied with a short textual description that describes its semantic meaning (Table 1). We also assume that a small amount of example slot values are given, following Shah et al. (2019).

Our goal is to build a slot filling model that performs well on a new target domain with unseen slot types. Our training data consists of utterances from N source domains $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N$. Each domain $\mathcal{D}_i$ is associated with a set of predefined slot types S_i. At test time, utterances are drawn from a new domain $\mathcal{D}_{N+1}$. The new domain contains both seen and unseen slot types from the source domain. For example, in the SNIPS dataset (Coucke et al., 2018), domains "GetWeather" and "BookRestaurant" both have a slot type called "city", while "condition_temperature" only appears in the "GetWeather" domain.

predicate-argument structures are represented as QA pairs.

3 Methodology

In this section, we describe our framework for Question Answer-driven Slot Filling (QASF). The framework consists of (1) a *question generation strategy* that turns slot descriptions into natural language questions based on linguistic rules; (2) a generic span-extraction-based *question answering* model; (3) an intermediate pretraining stage with generated synthetic QA pairs from unlabeled conversations, which is before task-specific training.

3.1 Question Generation Strategy

To benefit from both language model pretraining and QA supervision, we design a question generation strategy to turn slot descriptions into natural questions. During this process, a considerable amount of knowledge and semantic information is encoded (Heilman, 2011). A generated question consists of a WH word and a normalized slot description following the template below:

`WH_word` is `slot_description` ?

Generating `WH_word` We draw insights from the literature on automatic question generation. Heilman and Smith (2010) propose to use linguistically motivated rules. In their more general case of question generation from the sentence, answer phrases can be noun phrases (`NP`), prepositional phrases (`PP`), or subordinate clauses (`SBAR`). Complicated rules are designed with help from superTagger (Ciaramita and Altun, 2006).

For our spoken language understanding (SLU) tasks, slot fillers are mostly noun phrases[2]. Therefore, we design a simpler set of conditions based on named entity types and part-of-speech (POS) tags. For each slot type, we sample 10 (utterance, slot value) examples from the validation set. Then we run a NER and a POS tagging model [3] to obtain entity types and POS tags for each of the sampled answer spans. Finally, we select `WH_word` based on a set of rules described in Table 6 in Appendix.

Generating `slot_description` Instead of directly adding a raw description phrase in the question template, we *normalize* the phrase with the

[2] around 90% cases in the SNIPS dataset.

[3] Provided by spaCy: `https://spacy.io/`

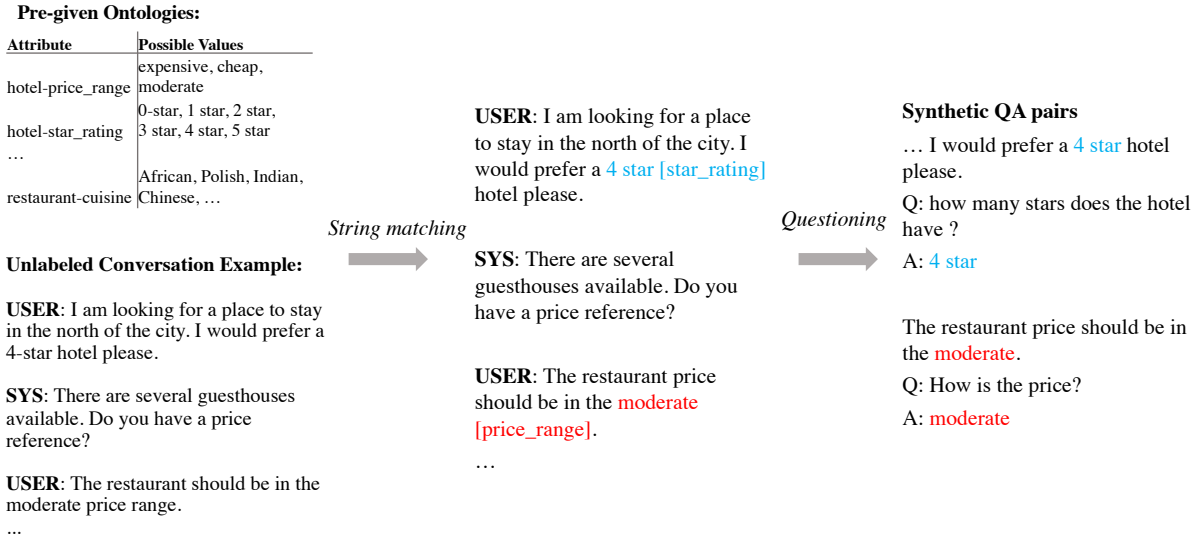


Figure 2: Obtaining weakly supervised synthetic QA pairs for pretraining. Given an ontology and unlabeled utterances, we generate synthetic QA pairs with weak supervision by matching values of slots against the utterances.

Slot	Raw Description	Our Question
playlist_owner	owner	who's the owner?
object_select	object select	which object to select?
best_rating	points in total	how many points in total?
num_book_people	number of people for booking	how many people for booking?

Table 1: Examples of generated questions.

following simple rule: *If the description is of the format "A of B", where both A and B are noun phrases (NP), we only keep B in the phrase if the WH_word is "How long" or "How many"*. Examples of generated questions for corresponding slots are presented in Table 1. Compared to slot descriptions, our questions are more precise and can encode more semantic information.

3.2 Question Answering Model

We use BERT (Devlin et al., 2019) as our base model for jointly encoding the question and utterance. Input sequences for the model share a standard BERT-style format: *[CLS] <question> [SEP] <utterance> [SEP]*, where [CLS] is BERT's special classification token and [SEP] is the special token to denote separation. Let $\mathbf{e}_{1:M}$ be the token-level output representation from the BERT encoder,

$$\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_M = \texttt{BERT}(x_1, x_2, ..., x_M) \quad (1)$$

where $x_{1:M}$ are the input tokens.

Then the model predicts answer spans with two binary classifiers on top of the BERT outputs $\mathbf{e}_{1:M}$. The two classifiers are trained to predict whether each token is the start or the end of an answer span, respectively,

$$P_{\text{s}}(i \mid i \in 1 \dots M) = \texttt{softmax}(\mathbf{e}_i \mathbf{W}_{\text{s}})$$
$$P_{\text{e}}(i \mid i \in 1 \dots M) = \texttt{softmax}(\mathbf{e}_i \mathbf{W}_{\text{e}})$$

For negative examples, where a question has no answer spans in the utterance, we map the start and end token both to the [CLS] token. During training, we minimize the negative log-likelihood loss. All parameters are updated. During inference, predicting slot filler spans is more complex because there could be *several* or *no* spans to be extracted for each slot type. We first enumerate all possible spans and only keeping spans/answers satisfying certain constraints (Appendix Section B) as fillers.

3.3 Pretraining with Weak Supervision

Pretrained masked language models do not have the capability of question answering before being fine-tuned on task-specific data. We hypothesize that adding a pretraining step with synthetic QA pairs before fine-tuning can contribute to models' understanding of interactions between question and utterance. For example, improvements have been reported by QAMR (He et al., 2020) on SRL and textual entailment (TE). Previous researches (Wu et al., 2020; Gao et al., 2020) have used crowd-sourced QA pairs, but typically the improvement margin is not significant (Wu et al., 2020) when the task-specific data is in a different domain (SQuAD v.s. newswire). Therefore we introduce a method of collecting relevant and distantly supervised QA pairs and investigate their influences in pretraining. More specifically, we draw insights from

Mintz et al. (2009) for creating a weakly supervised dataset. Figure 2 illustrates the process. Given an *ontology or database* of slot types and all possible values for each slot type, we find all utterances containing those value strings in a large set of **unlabeled conversations**. For example, in Figure 2, for the "hotel_price_range" slot, there are three possible values "expensive", "cheap" and "moderate" in the ontology. We then form question-answer-utterance triples using the question generation strategy proposed in Section 3.1.

To obtain the pre-defined ontology and unlabeled conversations, we use MultiWOZ 2.2 (Zang et al., 2020), which is an improved version of MultiWOZ (Budzianowski et al., 2018). We do not use annotations in the dataset such as the (changes of) states in the conversations and we treat each utterance independently. We remove slot types that exist in the task-specific training/test data (i.e., SNIPS and TOP) from the ontology and end up with 67,370 QA examples for pretraining.

4 Experiments

4.1 Datasets and Baselines

SNIPS (Coucke et al., 2018) is an SLU dataset consisting of crowdsourced user utterances with 39 slots across 7 domains – "AddToPlaylist" (ATP), "BookRestaurant" (BR), "GetWeather" (GW), "PlayMusic" (PM), "RateBook" (RB), "SearchCreativeWork" (SCW), "FindScreeningEvent" (FSE). It has around 2000 training instances per domain. The slot types of each domain do not overlap with each other. Following previous work (Shah et al., 2019; Liu et al., 2020), we use this dataset to evaluate zero-shot cross-domain transfer learning – train on all training instances from domains other than D_i, and test exclusively on D_i, for $i = 1, \ldots, 7$.
TOP (Gupta et al., 2018) is a task-oriented utterance parsing dataset. It is based on a hierarchical annotation scheme for annotating utterances with nested intents and slots. Each slot type also comes with a description. In our setup, we train on all seven domains of SNIPS as well as varying amounts of training data from the TOP training set (0, 20, and 50 examples), and use the TOP test set as an out-of-distribution domain for evaluation. We report span-level F1 (micro-average).

We compare our method against a number of representative baselines. Concept Tagger (CT) (Bapna et al., 2017) is a slot-filling framework that directly uses *original* slot descriptions to generalize to unseen slot types. Robust Zero-shot Tagger (RZT) (Shah et al., 2019) is an extension of CT, which incorporates example values of slots to improve the robustness of the model's zero-shot capability. Coach (Liu et al., 2020) is a coarse-to-fine model for slot-filling. It also encodes raw slot descriptions. We also include a Zero-Shot BERT Tagger (ZSBT) based on BERT (Devlin et al., 2019) as an additional baseline. ZSBT directly encodes raw slot descriptions and utterances and predicts a tag (B, I, or O) for each token in the utterance.

4.2 Results and Analysis

We report F-1 of the baselines and our model on each target domain test set of **SNIPS** as well as average F-1 across domains. All models are trained on the other six domains for each target domain. As shown in Table 2, our QA-driven slot filling framework (QASF) significantly outperforms all baselines in five of the seven domains, with slightly lower performance on BookRestaurant than ZSBT, and lower performance on FindScreeningEvent than Coach. The average F-1 of QASF is around 7% higher than the prior published state-of-the-art Coach model, and about 2% higher than the Zero-shot BERT Tagger baseline. Adding the intermediate pre-training stage on weakly supervised data further improves performance on top of QASF in six of the seven domains except for AddToPlaylist. On average, adding pre-training improves over QASF by 2.9% F-1. The zero-shot performance of all models are relatively worse on PlayMusic, RateBook and FindScreeningEvent. A more detailed discussion is in Appendix Section C.

Table 3 summarizes **TOP test** results: (1) In both the zero-shot and few-shot settings, our QASF outperforms ZSBT, with a bigger improvement on the zero-shot setting. (2) Pretraining on the weakly supervised QA pairs helps more in the zero-shot setting than in the few-shot setting, with a 20% relative improvement. This shows that QASF (w/ pretraining) is more robust to the domain shift when there is no target domain training data.

Impact of QG strategy and pretraining To understand the influence of question generation and impact of pretraining with synthetic QA pairs, we perform ablation studies of both components on the SNIPS dataset. The table below shows ablation results (F-1). "w/o QG" refers to a model trained with *raw* slot descriptions and utterances.

Firstly, the question generation strategy consis-

	ATP	BR	GW	PM	RB	SCW	FSE	Average F-1
CT (Bapna et al., 2017)	38.82	27.54	46.45	32.86	14.54	39.79	13.83	30.55
RZT (Shah et al., 2019)	42.77	30.68	50.28	33.12	16.43	44.45	12.25	32.85
Coach (Liu et al., 2020)	50.90	34.01	50.47	32.01	22.06	46.65	25.63	37.39
ZSBT$^{\text{BERT}}$ (our baseline)	55.78	**49.34**	56.58	28.35	27.09	57.61	20.50	42.18
QASF$^{\text{BERT}}$ (ours)	**59.29**	43.13	59.02	33.62	33.34	59.90	22.83	44.45
w/ pre-training on WS	57.57	48.75	**61.27***	**38.54***	**36.51****	**60.82**	**27.72****	47.31

Table 2: Experimental results (F-1) on SNIPS dataset. * indicates statistical significance ($p < 0.05$), **: $p < 0.01$.

	Zero-shot	Few-shot (20)	Few-shot (50)
Random NE	1.34	-	-
ZSBT	8.82	37.60	42.73
QASF (ours)	10.27	36.86	46.49
w/ pre-training on WS	12.35	39.78	47.91

Table 3: Evaluation results on TOP test. Models trained on SNIPS, and varying amount of utterances of TOP train – zero-shot, 20-shot (1%), 50-shot (2.5%).

w/o pretraining			w/ pretraining		
QASF	w/o QG	Δ (F1)	QASF	w/o QG	Δ (F1)
44.45	41.97	+5.91%	47.31	43.09	+9.79%

Table 4: Ablation Study

tently helps, with a 2.48% F-1 gain in "w/o pretraining" and a 4.22% F-1 gain in "w/ pretraining". Secondly, the pretrained representations from additional weakly supervised data improve F-1 by 2.86% in "w/ QG" and 1.12% in "w/o QG". More interestingly, the gain from the questioning strategy is larger when combined with the pretraining (9.8% as compared to 5.9%). This demonstrates that synthetic QA pairs are also helping with getting better QA-aware representations before fine-tuning on the task-specific data for slot-filling.

4.3 Error Analysis

We further conduct manual error analysis on the models' predictions on SNIPS. We find that there are several sources where the errors are from:

The variance between the source and target domains. Sometimes even slot types of the same name refer to different kinds of objects in different domains. For example, slot type "object_type" in the "RateBook" domain refers to object types like textbook, essay and novel; while in the "FindScreeningEvent", it refers to event types like movie times/schedules. In the two domains, they have the same raw descriptions. In the table below, we show the performance of models on utterances with "object_type" and "object_name" spans (according to gold annotations). We can see that the performances on these special slots are significantly

	ZSBT	QASF	QASF (w/ pretraining)
Averaged F-1	17.43	22.26	24.29

lower than the general average on all the examples (40–50%). But still, the questioning strategy helps improve the transferring of semantic information.

Plus, the variance in semantic meaning between slot types in SNIPS and TOPS is even larger. For slots like "location_modifier", "road_condition", there are no semantic similar slots in SNIPS or pretraining dataset, which results in low performance. Having more specific/detailed slot descriptions and use them in the question generation would help further (Brown et al., 2020; Du and Cardie, 2020).

Annotation artifacts of SNIPS dataset and sparsity of vocabulary for certain slot types. Our QASF framework does not perform well on the target domain "BookRestaurant", thus we take a close look at it. We find that there are only 25 possible values in total for slot restaurant_type, over **51%** of them are of a single token "restaurant" (Table below). A very simple approach (assigning type "restaurant_type" to all tokens "restaurant" can obtain decent performance). This does not happen for

Slot Value	"restaurant"	"bar"	"pub"	"brasserie"
Proportion	51.81%	7.26%	6.60%	6.23%

other slot types in BookRestaurant (e.g., cuisine, restaurant_name). The possible values are more diverse and the distribution is more balanced.

5 Conclusion

We propose a QA-driven method with weakly supervised pretraining for zero-shot slot filling. Our experimental results and analyses demonstrate the benefits of QA-formulation, especially in the setting with synthetic QA pairs pretraining.

Acknowledgments

We thank Kenton Lee and Emily Pitler, and anonymous reviewers for their constructive suggestions.

References

Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. In *Proc. Interspeech 2017*, pages 2476–2480.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 594–602.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89, Online. Association for Computational Linguistics.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. QuASE: Question-answer driven sentence encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8743–8758, Online. Association for Computational Linguistics.

Luheng He, M. Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*.

Michael Heilman. 2011. *Automatic factual question generation from text*. Ph.D. thesis, Ph. D. thesis, Carnegie Mellon University.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 19–25, Online. Association for Computational Linguistics.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Julian Michael, Gabriel Stanovsky, Luheng He, I. Dagan, and Luke Zettlemoyer. 2017. Crowdsourcing question-answer meaning representations. *ArXiv*, abs/1711.05885:560–568.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tür. 2020. Language model is all you need: Natural language understanding as question answering. *arXiv preprint arXiv:2011.03023*.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2011. Annotating and learning event durations in text. *Computational Linguistics*, 37(4):727–752.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490, Florence, Italy. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Steve Young. 2002. Talking to machines (statistically speaking). In *Seventh International Conference on Spoken Language Processing*.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

A Details of Question Generation

The part-of-speech tagset is based on the Universal Dependencies scheme [4]. The named entity labels are based on OntoNotes 5.0 (Weischedel et al., 2013). In Table 6, we describe the set of rules for the selection of `WH_word`.

B Inference Constraints

At inference time, predicting the slot filler spans is more complex – for each slot type, as there can be *several* or *no* spans to be extracted. After the output layer, we have the probability of each token x_i being the start ($P_{\text{s}}(i)$) and end ($P_{\text{e}}(i)$) of the span.

We harvest all the valid candidate spans for each slot type with the following heuristics:

1. Enumerate all possible combinations of start offset (start) and end offset (end) of the spans ($\frac{M(M-1)}{2}$ candidates in total);

2. Eliminate the spans not satisfying the constraints: (1) start and end token must be within the utterance; (2) the length of the span should be shorter than a maximum length constraint; (3) spans should have a larger probability than the probability of "no answer" (which is represented with [CLS] token), namely,

$$P_{\text{s}}(\text{start}) > P_{\text{s}}([\text{CLS}]), P_{\text{e}}(\text{end}) > P_{\text{e}}([\text{CLS}])$$

C Further Analysis and Discussions

We conduct further analysis to understand how and why the models are effective.

C.1 Impact of question generation strategy and pretraining

Table 7 shows full ablation results.

C.2 Analysis on Seen versus Unseen Slots

To understand the transferring capability of our models, we further split the SNIPS test for each target domain into "seen" and "unseen" slots. An example is categorized into "unseen" as long as there is an unseen slot (i.e., the slot does not exist in the remaining six source domains in its gold annotation.) Otherwise, it counts as "seen". A full list of unseen slots for each target domain can be found in the Appendix.

[4] universaldependencies.org/u/pos/

As is shown in Table 5, we can see that (1) both ZSBT baseline and our models perform better on the "seen" slots than the "unseen" ones – the numbers substantially drop on the "unseen" slots. This proves that transferring from the source domains to the unseen slots in the target domain is a hard problem. (2) On the portion of examples with "seen" slots, our best model outperforms ZSBT with around a 2% margin. (3) On the "unseen" portion of examples, the margin is larger – our QASF and pretraining step help improve the performance more (over 4%). The second and third observation together demonstrates that the questioning strategy help improve the model's capability of transferring between related but not exactly same slot types (e.g., "object_name" and "entity_name").

	Seen	Unseen
ZSBT	54.75	37.41
QASF	56.79	39.99
w/ pretraining	56.23	41.73

Table 5: Averaged F-1 scores over all target domains of SNIPS dataset (for "unseen" and seen "slots").

D Hyper-parameters and Training Details

We use the uncased version of the BERT-base (Devlin et al., 2019) model for QA finetuning and pretraining. The model is fine-tuned for 5 epochs with a starting learning rate of 3e-5 on the SNIPS dataset. The model is pretrained for 5 epochs with a starting learning rate of 5e-7 on the synthetic QA dataset. Our implementations are based on https://github.com/google-research/bert/blob/master/run_squad.py

WH_word	Conditions	Answer Examples
How long	The answer phrase is modified by a cardinal number (CARDINAL) or quantifier phrase (QUANTITY) whose object is a temporal unit, as is defined in (Pan et al., 2011), i.e., second/minute/hour/day/week/month/year/decade/century.	2 nights
How many	The answer phrase is modified by a cardinal number (CARDINAL) or quantifier phrase (QUANTITY) and the object is not a temporal unit.	2 stars, 3 tickets
How	adjective ADJ	moderate, expensive
When	The answer phrase's head word is tagged DATE or TIME	1:30 PM, 1999
Who	The answer phrase's head word is tagged PERSON or is a personal pronoun PRON (I, he, herself, them, etc.)	mother, Dr. Williams
Where	The answer phrase is a prepositional phrase whose object is tagged GPE or LOC, whose preposition is one of the following: on, in, at, over, to	amc theaters, fort point san francisco, east, west, ...
Which	The answer phrase is a determiner DT (this, that) or an ordinal ORDINAL	this, first, current, last
What	all other cases	

Table 6: Strategy for Generating the WH_word (question phrase).

	ATP	BR	GW	PM	RB	SCW	FSE	Average F1
w/o pretraining								
QASF	59.29	43.13	59.02	33.62	33.34	59.90	22.83	44.45
w/o question	55.30	46.71	53.06	35.79	25.28	59.77	17.85	41.97
w/ pretraining								
QASF	57.57	48.75	61.27	38.54	36.51	60.82	27.72	47.31
w/o question	56.11	42.42	55.70	33.07	33.13	60.03	21.20	43.09

Table 7: Full ablation analysis on SNIPS dataset.

E Schema of SNIPS dataset

Domain	All Slots	Unseen Slots
AddToPlaylist (ATP)	entity_name playlist_owner playlist artist music_item	entity_name playlist_owner
BookRestaurant (BR)	restaurant_type served_dish restaurant_name party_size_description cuisine party_size_number timerange facility poi state city country sort spatial_relation	restaurant_type served_dish restaurant_name party_size_description cuisine party_size_number timerange facility poi
GetWeather (GW)	timerange current_location condition_description geographic_poi condition_temperature country state city spatial_relation	timerange current_location condition_description geographic_poi condition_temperature
PlayMusic (PM)	year genre service album track sort music_item artist playlist	year genre service album track
RateBook (RB)	object_select rating_value best_rating rating_unit object_part_of_series_type object_type object_name	object_select rating_value best_rating rating_unit object_part_of_series_type
SearchCreativeWork (SCW)	object_type object_name	-
FindScreeningEvent (FSE)	object_location_type movie_name movie_type timerange location_name object_type spatial_relation	object_location_type movie_name movie_type timerange location_name

Table 8: Schema of SNIPS dataset

F Question Templates for SNIPS

Domain	Slot	Slot Name	Natural Question
AddToPlaylist	music_item	music item	what's the music item?
AddToPlaylist	playlist_owner	owner	who's the owner?
AddToPlaylist	entity_name	entity name	what's the entity name?
AddToPlaylist	playlist	playlist	what's the playlist?
AddToPlaylist	artist	artist	who's the artist?
BookRestaurant	city	city	what's the city?
BookRestaurant	facility	facility	what's the facility?
BookRestaurant	timeRange	time range	when's the time range?
BookRestaurant	restaurant_name	restaurant name	what's the name?
BookRestaurant	country	country	what's the country?
BookRestaurant	cuisine	cuisine	what's the cuisine?
BookRestaurant	restaurant_type	restaurant type	what's the restaurant type?
BookRestaurant	served_dish	served dish	what's the served dish?
BookRestaurant	party_size_number	number	how many people?
BookRestaurant	poi	position	where's the location?
BookRestaurant	sort	type	what's the type?
BookRestaurant	spatial_relation	spatial relation	what's the spatial relation?
BookRestaurant	state	location	what's the state?
BookRestaurant	party_size_description	person	who are the persons?
GetWeather	city	city	what's the city?
GetWeather	state	location	what's the state?
GetWeather	timeRange	time range	when's the time range?
GetWeather	current_location	current location	what's the current location?
GetWeather	country	country	what's the country?
GetWeather	spatial_relation	spatial relation	what's the spatial relation?
GetWeather	geographic_poi	geographic position	where's the location?
GetWeather	condition_temperature	temperature	how is the temperature?
GetWeather	condition_description	weather	how is the weather?
PlayMusic	genre	genre	what's the genre?
PlayMusic	music_item	music item	what's the music item?
PlayMusic	service	service	what's the service
PlayMusic	year	year	when's the year?
PlayMusic	playlist	playlist	what's the playlist?
PlayMusic	album	album	what's the album?
PlayMusic	sort	type	what's the type?
PlayMusic	track	track	what's the track?
PlayMusic	artist	artist	who's the artist?
RateBook	object_part_of_series_type	series	what's the series?
RateBook	object_select	this current	which to select?
RateBook	rating_value	rating value	how many rating value?
RateBook	object_name	object name	what's the object name?
RateBook	object_type	object type	what's the object type?
RateBook	rating_unit	rating unit	what's the rating unit?
RateBook	best_rating	best rating	how many rating points in total?
SearchCreativeWork	object_name	object name	what's the object name?
SearchCreativeWork	object_type	object type	what's the object type?
SearchScreeningEvent	timeRange	time range	when's the time range?
SearchScreeningEvent	movie_type	movie type	what's the movie type?
SearchScreeningEvent	object_location_type	location type	what's the location type?
SearchScreeningEvent	object_type	object type	what's the object type?
SearchScreeningEvent	location_name	location name	where's the location name?
SearchScreeningEvent	spatial_relation	spatial relation	what's the spatial relation?
SearchScreeningEvent	movie_name	movie name	what's the movie name?

Table 9: Question Templates for SNIPS

Domain-Adaptive Pretraining Methods for Dialogue Understanding

Han Wu[1], **Kun Xu**[2], **Linfeng Song**[2], **Lifeng Jin**[2], **Haisong Zhang**[2], **Linqi Song**[1]
[1]Department of Computer Science, City University of Hong Kong
[2]Tencent AI Lab
{hanwu32-c}@my.cityu.edu.hk
{kxkunxu,lfsong,lifengjin, hansongzhang}@tencent.com
{linqi.song}@cityu.edu.hk

Abstract

Language models like BERT and SpanBERT pretrained on open-domain data have obtained impressive gains on various NLP tasks. In this paper, we probe the effectiveness of domain-adaptive pretraining objectives on downstream tasks. In particular, three objectives, including a novel objective focusing on modeling predicate-argument relations, are evaluated on two challenging dialogue understanding tasks. Experimental results demonstrate that domain-adaptive pretraining with proper objectives can significantly improve the performance of a strong baseline on these tasks, achieving the new state-of-the-art performances.

1 Introduction

Recent advances in pretraining methods (Devlin et al., 2019; Joshi et al., 2020; Yang et al., 2019) have achieved promising results on various natural language processing (NLP) tasks, including natural language understanding, text generation and question anwsering (Liu et al., 2019; Song et al., 2019; Reddy et al., 2019). In order to acquire general linguistic and semantic knowledge, these pretraining methods are usually performed on open-domain corpus, like Wikipedia and BooksCorpus. In light of the success from open-domain pretraining, a further question is naturally raised: whether downstream tasks can also benefit from domain-adaptive pretraining?

To answer this question, later work (Baevski et al., 2019; Gururangan et al., 2020) has demonstrated that continued pretraining on the unlabeled data in the target domain can further contribute to the corresponding downstream task. However, these studies are dependent on additional data that can be unavailable in certain scenarios, and they only evaluated on easy downstream tasks. For instance, Gururangan et al. (2020) perform continued pretraining with masked language modeling loss on several relevant domains, and they obtain improvements on eight well-studied classification tasks, which are too simple to exhibit the strength of continued domain-adaptive pretraining. Besides, it is still unclear which pretraining objective is the most effective for each downstream task.

In this work, we give a deeper analysis on how various domain-adaptive pretraining methods can help downstream tasks. Specifically, we continuously pretrain a BERT model (Devlin et al., 2019) with three different kinds of unsupervised pretraining objectives on the domain-specific training set of each target task. Two of them are Masked Language Model (MLM) (Gururangan et al., 2020) and Span Boundary Objective (SBO) (Joshi et al., 2020), both objectives have been explored in previous work. In addition, a novel pretraining objective, namely Perturbation Masking Objective (PMO), is proposed to better learn the correlation between arguments and predicates. After domain-adaptive pretraining, the adapted BERT is then tested on dialogue understanding tasks to probe the effectiveness of different pretraining objectives.

We evaluate on two challenging tasks that focus on dialogue understanding, i.e. Conversational Semantic Role labeling (CSRL) and Spoken Language Understanding (SLU). CSRL (Xu et al., 2020, 2021) was recently proposed by extending standard semantic role labeling (SRL) (Palmer et al., 2010) with cross-utterance relations, which otherwise require coreference and anaphora resolution for being recognized. We follow previous work to consider this task as sequence labeling. On the other hand, SLU includes intent detection and slot filling. To facilitate domain-adaptive pretraining, we only use the training set of each downstream task. In this way, the usefulness of each pretraining objective can be more accurately examined, as no additional data is used.

Experimental results show that domain-adaptive

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 665–669
August 1–6, 2021. ©2021 Association for Computational Linguistics

pretraining significantly helps both tasks. Besides, our novel objective achieves better performances than the existing ones, shedding more lights for future work on pretraining.

2 Tasks

Conversational Semantic Role Labeling. Xu et al. (2021) first proposed the CSRL task, which extends standard SRL by explicitly annotating other cross-turn predicate-argument structures inside a conversation. Compared with newswire documents, human conversations tend to have more ellipsis and anaphora situations, causing more problems for standard NLU methods. Their motivation is that most dropped or referred components in the latest dialogue turn can actually be found in the dialogue history. As the result, CSRL allows arguments to be in different utterances as the predicate, while SRL can only work on each single utterance. Comparing with standard SRL, CSRL can be more challenging due to the long-range dependencies. Similar to SRL, we view CSRL as a sequence labeling problem, where the goal is to label each token with a semantic role.

Spoken Language Understanding. Proposed by Zhu et al. (2020), the SLU task consists of two key components, i.e., intent detection and slot filling. Given a dialogue utterance, the goal is to predict its intents and to detect pre-defined slots, respectively. We treat them as sentence-level classification and sequence labeling, respectively.

3 Domain-Adaptive Pretraining Objectives

While previous works have shown the benefit of continued pretraining on domain-specific unlabeled data (e.g., Lee et al. (2020); Gururangan et al. (2020)), these methods only adopt the Masked Language Model (MLM) objective to train an adaptive language model on a single domain. It is not clear how the benefit of continued pretraining may vary with factors like the objective function.

In this paper, we use the dialogue understanding task as a testbed to investigate the impact of three pre-training objectives to the overall performance. In particular, we explore the MLM (Devlin et al., 2019) and Span Boundary Objective (SBO) (Joshi et al., 2020) , and introduce a new objective, namely Perturbation Masking Objective (PMO), which is more fit for the dialogue NLU task.

3.1 Masked Language Model Objective

Masked Language Model (MLM) is the task of predicting missing tokens in a sequence from their placeholders. Specifically, given a sequence of tokens $X = (x_1, x_2, .., x_n)$, a subset of tokens $Y \subseteq X$ is sampled and substituted with a different set of tokens. In BERT's implementation, Y accounts for 15% of the tokens in X; of those, 80% are replaced with [MASK], 10% are replaced with a random token (according to the unigram distribution), and 10% are kept unchanged. Formally, the contextual vector of input tokens X is denoted as $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n)$. The task is to predict the original tokens in Y from the modified input and the objective function is:

$$\mathcal{L}_{MLM} = -\frac{1}{|Y|}\sum_{t=1}^{|Y|}\log p(x_t|\boldsymbol{h}_t;\boldsymbol{\theta})$$

where $|Y|$ is the number of masked tokens, and $\boldsymbol{\theta}$ represents the model parameters.

3.2 Span Boundary Objective

In many NLP tasks such as the dialogue understanding, it usually involves reasoning about relationships between two or more spans of text. Previous works (Joshi et al., 2020) have shown that SpanBERT is superior to BERT in learning span representations, which significantly improves the performance on those tasks. Conceptually, the differences between these two models are two folds.

Firstly, different with BERT that independently selects the masked token in Y, SpanBERT define Y by randomly selecting *contiguous spans*. In particular, SpanBERT first selects a subset $Y \subseteq X$ by iteratively sampling spans until masking 15% tokens[1]. Then, it randomly (uniformly) selects the starting point for the span to be masked.

Secondly, SpanBERT additionally introduces a span boundary objective that involves predicting each token of a masked span using only the representations of the observed tokens at the boundaries. For a masked span of tokens $(x_s, ..., x_e) \in Y$, where (s, e) are the start and end positions of the span, it represents each token in the span using the boundary vectors and the position embedding:

$$\boldsymbol{y_i} = f(\boldsymbol{h}_{s-1}, \boldsymbol{h}_{e+1}, \boldsymbol{p}_{i-s+1})$$

where p_i marks relative positions of span token x_i with respect to the left boundary token x_{s-1},

[1]The length of each span is sampled from the geometric distribution $l \sim Geo(p)$, with $p = 0.2$.

and $f(\cdot)$ is a 2-layer MLP with GeLU activations and layer normalization. SpanBERT sums the loss from both the regular MLM and the span boundary objectives for each token in the masked span:

$$\mathcal{L}_{SBO} = -\frac{1}{|Y|}\sum_{t=1}^{|Y|}\log p(x_t|\boldsymbol{y}_t;\boldsymbol{\theta})$$

3.3 Perturbation Masking Objective

In dialogue understanding tasks like CSRL, the major goal is to capture the semantic information such as the correlation between arguments and predicate. However, for the sake of generalization, existing pretraining models do not consider the semantic information of a word and also not assess the impact of predicate has on the prediction of arguments in their objectives. To address this, we propose to use the perturbation masking technique (Wu et al., 2020) to explicitly measure the correlation between arguments and predicate and further introduce that into our objective.

The perturbation masking is originally proposed to assess the impact one word has on the prediction of another in MLM. In particular, given a list of tokens X, we first use a pretrained language model $\mathbf{M}$ to map each x_i into a contextualized representation $H(X)_i$. Then, we use a two-stage approach to capture the impact word x_j has on the prediction of another word x_i. First, we replace x_i with the [MASK] token and feed the new sequence $X\backslash\{x_i\}$ into $\mathbf{M}$. We use $H(X\backslash\{x_i\})_i$ to denote the representation of x_i. To calculate the impact $x_j \in x\backslash\{x_i\}$ has on $H(X)_i$, we further mask out x_j to obtain the second corrupted sequence $X\backslash\{x_i, x_j\}$. Similarly, $H(X\backslash\{x_i, x_j\})_i$ denotes the new representation of token x_i. We define the the impact function as: $f(x_i, x_j) = d(H(X\backslash\{x_i\})_i, H(X\backslash\{x_i, x_j\})_i)$, where d is the distance metric that captures the difference between two vectors. In experiments, we use the Euclidean distance as the distance metric.

Since our goal is to better learn the correlation between arguments and predicate, we introduce a perturbation masking objective that maximizes the impact of predicate on the prediction of argument span:

$$\mathcal{L}_{PMO} = -\frac{1}{|Y|}\sum_{t=1}^{|Y|} -f(x_t, \{x_{p_0}, ..., x_{p_{m-1}}\})_i$$

where p_0,... p_{m-1} are m predicates that occur in the sentence. In practice, we first follow the SpanBERT to sample a subset of contiguous span texts and perform masking (i.e., span masking) on them. Then, we select verbs from X as predicates and perform perturbation masking on those predicates.

4 Experiments

We evaluate pretraining objectives on three datasets, DuConv, NewsDialog[2] and CrossWOZ. The former two datasets are annotated by Xu et al. (2021) for the CSRL task and the last one is provided by Zhu et al. (2020) for the SLU task.

Duconv is a Chinese knowledge-driven dialogue dataset, focusing on the domain of movies and stars. NewsDialog is a dataset collected in a way that follows the setting for constructing general open-domain dialogues: two participants engage in chitchat, and during the conversation, the topic is allowed to change naturally. Xu et al. (2021) annotates 3K dialogue sessions of DuConv to train their CSRL parser, and directly test on 200 annotated dialogue sessions of NewsDialog. CrossWOZ is a Chinese Wizard-of-Oz task-oriented dataset, including 6K dialogue sessions and 102K utterances on five domains.

Since the state-of-the-art models on these tasks are all developed based on BERT, we use the same model architectures but just replace the BERT base with our domain-adaptive pretrained BERT. Notice that, we also experiment with other pretrained language models such as RoBERTa and XLNet. We observed similar results but here we only report the results based on BERT due to the space limitation.

In particular, we perform the domain-adaptive pretraining on CSRL task using all dialogue sessions of training set in DuConv (Wu et al., 2019) and NewsDialog (Wang et al., 2021), which includes 26K and 20K sessions, respectively; on the SLU task, we use the whole CrossWOZ training dataset.

The hyper-parameters used in our model are listed as follows. The network parameters of our model are initialized using the pretrained language model. The batch size is set to 128. We use Adam (Kingma and Ba, 2015) with learning rate 5e-5 to update parameters.

Results and Discussion. On the CSRL task, we follow Xu et al. (2021) to use the micro-averaged F1 over the (*predicate*, *argument*, *label*) tuples. Specifically, we calculate F1 over all arguments

[2]We obtain the CSRL annotations on DuConv and NewsDialog directly from the author of Xu et al. (2021).

Pretraining Strategy	DuConv			NewsDialog			CrossWOZ		
	$F1_{all}$	$F1_{cross}$	$F1_{intra}$	$F1_{all}$	$F1_{cross}$	$F1_{intra}$	$F1_{intent}$	$F1_{slot}$	$F1_{all}$
No Pretraining	88.16	83.74	88.71	76.81	53.61	79.97	95.67	95.13	95.34
MLM	88.56	84.37	88.97	76.93	53.43	80.15	95.85	95.47	95.62
MLM + SBO	88.73	84.49	89.23	78.10	56.21	80.85	96.17	95.54	95.78
MLM + PMO	89.10	85.26	89.52	79.68	56.19	81.79	96.40	95.79	96.17
MLM + SBO + PMO	89.21	85.98	89.79	80.01	56.20	82.78	96.48	96.03	96.21
w/ NP Sampling ($\alpha = 50$)	89.34	86.12	89.99	81.32	56.67	83.14	96.81	96.52	96.70
w/ NP Sampling ($\alpha = 80$)	**89.97**	**86.68**	**90.31**	**81.90**	**56.56**	**84.56**	**96.97**	**96.87**	**96.93**

Table 1: Evaluation on the DuConv, NewsDialog and CrossWOZ. α is the ratio of sampling from noun phrases.

(referred as $F1_{all}$) and those in the same and different dialogue turns as predicates (referred as $F1_{intra}$ and $F1_{cross}$). On the SLU task, we report results on $F1_{intent}$, $F1_{slot}$ and $F1_{all}$. Table 1 summarizes the results. The first row shows the performance of existing state-of-the-art models without domain-adaptive pretraining on each dataset. We can see that on two tasks, existing models could benefit from the domain-adaptive pretraining, achieving new state-of-the-art performance on these datasets.

Let us first look at the CSRL task. Pretraining with MLM objective could slightly improve the performance by 0.4 and 0.12 in terms of $F1_{all}$ on DuConv and NewsDialog, respectively. By additionally considering the span boundary objective, the overall performance especially $F1_{cross}$ could be further improved by at least 0.75 and 2.6, respectively. These results are expected since arguments in the CSRL task are usually spans and SBO is better than MLM in learning the span representation. We can also see that our proposed perturbation masking objective boosts the performance by a larger margin than SBO, indicating that learning correlations between arguments and predicates is more crucial to the NLU task. By summing three objectives, the CSRL model could achieve the best performance, significantly improving the baseline that without domain-adaptive pretraining by 1.05 and 3.2 $F1_{all}$ score, respectively.

From Table 1, we can see that similar findings are also observed on the SLU task. First of all, domain-adaptive pretraining on CrossWOZ could also improve the performance. Secondly, adding either SBO or PMO, the F1 scores on intent and slot could be further improved. Thirdly, the best performance is achieved when all three objectives are considered. However, we do not observe similar substantial gains on the SLU task as on the CSRL task. We think this is because the state-of-the-art performance on CrossWOZ is relatively high, but it is still impressive to achieve absolute 0.81, 0.90 and 0.87 points improvement in terms of $F1_{intent}$, $F1_{slot}$ and $F1_{all}$.

We also investigate the impact of span masking scheme to the overall performance. Recall that, in the span masking, we randomly sample the span length and a start position of the span. Joshi et al. (2020) showed that no significant performance gains are observed by using more linguistically-informed span masking strategies such as masking *Named Entities* or *Noun Phrases*. Specifically, they use the spaCy's[3] named entity recognizer and constituency parser to extract named entities and noun phrases, respectively. In this paper, we revisit these span masking scheme. Since there is no available constituency parser designed for the dialogue, we use an unsupervised grammar induction method (Jin and Schuler, 2020) to extract grammars from the training data. **Noun phrases** from Viterbi parse trees from different grammars are tallied without labels, resulting in a posterior distributions of the spans, which are used in our span sampling. As shown in Table 1, we find the best choice is to combine random sampling and noun phrases sampling, i.e., sampling from the noun phrases at $\alpha\%$ of the time and from a geometric distribution for the other (1 - $\alpha\%$). The performance on all three datasets coherently increases when more noun phrases are used in the span sampling.

5 Conclusion

In this paper, we probe the effectiveness of domain-adaptive pretraining on dialogue understanding tasks. Specifically, we study three domain-adaptive pretraining objectives, including a novel objective: *perturbation masking objective* on three NLU datasets. Experimental results show that domain-adaptive pretraining with proper objectives is a sim-

[3]https://spacy.io/

ple yet effective way to boost the dialogue understanding performance.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable and constructive comments. This work was supported in part by the City University of Hong Kong Teaching Development Grants 6000755.

References

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5360–5369, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Lifeng Jin and William Schuler. 2020. Grounded pcfg induction with images. In *AACL-IJCNLP*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In International Conference on Learning Representations.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Wei Liu, Lei Li, Zuying Huang, and Yinan Liu. 2019. Multi-lingual Wikipedia summarization and title generation on low resource corpus. In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pages 17–25, Varna, Bulgaria. INCOMA Ltd.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Xiaoyang Wang, Chen Li, Jianqiao Zhao, and Dong Yu. 2021. Naturalconv: A chinese dialogue dataset towards multi-turn topic-driven conversation. *arXiv preprint arXiv:2103.02548*.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic role labeling guided multi-turn dialogue rewriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639.

Kun Xu, Han Wu, Linfeng Song, Haisong Zhang, Linqi Song, and Dong Yu. 2021. Conversational semantic role labeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

Targeting the Benchmark: On Methodology in Current Natural Language Processing Research

David Schlangen
CoLabPotsdam / Computational Linguistics
Department of Linguistics, University of Potsdam, Germany
david.schlangen@uni-potsdam.de

Abstract

It has become a common pattern in our field: One group introduces a *language task*, exemplified by a *dataset*, which they argue is *challenging* enough to serve as a *benchmark*. They also provide a baseline *model* for it, which then soon is *improved* upon by other groups. Often, research efforts then move on, and the pattern repeats itself. What is typically left implicit is the argumentation for why this constitutes progress, and progress towards what. In this paper, I try to step back for a moment from this pattern and work out possible argumentations and their parts.

1 Introduction

The goal of any field of research is to make progress towards answering its foundational questions. To do so, a *methodology* is required that guides attempts at providing or improving answer proposals. In natural language processing, the object of study is human language, and any methodology for doing research in this field will need to have some contact with examples of this object. This contact has become more and more direct in the past decades, with samples of language becoming more directly the material from which proposals (in the form of statistical *models*) are derived. Recent years have seen an increase in the collection of samples specifically for the purpose of creating *benchmarks*, against which progress in devising models can be measured. It is this function of *benchmarking*, and its role in a progress-oriented methodology, that this paper aims to investigate.

Figure 1 illustrates the basic structure of a benchmarking methodology: A *language task* is devised that is a) restricted enough to be managable with current methods, and b) deemed challenging for the *capabilities* that it involves.[1] For this task, a *dataset* is collected, often via crowd sourcing, on which in turn models are trained and compared, using evaluation metrics defined together with the task. What can we learn by following such a methodology? Let's look at the components first and then at ways in which this methodology is, might, and perhaps should be used.

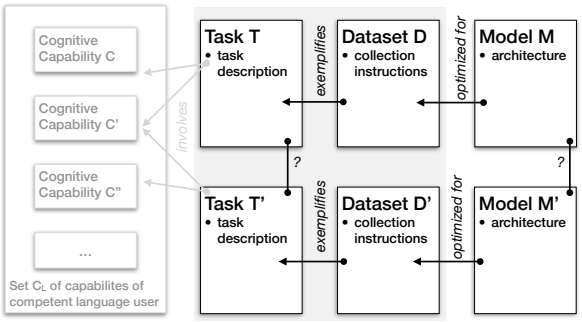


Figure 1: Relations between Research Objects in a Benchmark-Driven Methodology

2 What is a Language Benchmark?

2.1 What is a Benchmark?

In computing, a benchmark is *"a problem that has been designed to evaluate the performance of a system [which] is subjected to a known workload and the performance of the system against this workload is measured. Typically the purpose is to compare the measured performance with that of other systems that have been subject to the same benchmark test."* (Butterfield et al., 2016).

The use of this term in NLP is related: here, benchmark tasks are also specifically designed for evaluation; however, an important difference is that what is being evaluated is not a full system that has a separate main purpose, but rather an *algorithm* that is instantiated on the benchmark itself. I will discuss the consequences of this below.

This kind of evaluation of learning algorithm has a long tradition in the field of machine learning research.[2] In this field, a new algorithm would

[1] This figure is from (Schlangen, 2019), of which this is a shorter version developed in a somewhat different direction.

[2] For example, the UCI Machine Learning Repository has

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 670–674
August 1–6, 2021. ©2021 Association for Computational Linguistics

normally be tested on a large collection of datasets, possibly ranging from classifications of flowers to classification of credit records, with no assumption of any internal connection between the datasets. Again, NLP is different here, as all datasets represent facets of the same underlying phenomenon, language use.

I will argue that these two differences (life outside of benchmarking, and internal connection between tasks) are important, but understudied. But first we look at the notion of a *language task* in some more detail.

2.2 What is a Language Task?

A *language task* is a mapping between an *input space* and an *output* or *action space*, at least one of which contains natural language expressions. The mapping has to conform to a *task description*, which is typically given only informally, making reference to theoretical or pre-theoretical constructs external to the definition, such as "translation" or "is true of". I call this an *intensional description*. Typically, a task will also be specified *extensionally* through the provision of a *dataset* of examples of the mapping (that is, pairs of state and action). To collect such a dataset, the task description (e.g., "classification of entailment relations between sentence pairs") must be operationalised into a collection instruction ("please mark whether the situation that is well described by sentence A could normally also be described by sentence B").

3 How Can It be Evaluated?

3.1 Relation Task / Dataset

Given a task and a dataset, the first question to ask is how well the latter exemplifies the former. Investigating this is relatively straightforward. First, the dataset should be *verified*, which is to check whether the provided input/output pairs can indeed be judged correct relative to the task (in its intensional description). If the examples are collected specifically for the purpose of exemplifying the task, this is the process of controlling annotation, and standard methodologies exist (Artstein and Poesio, 2008; Pustejovsky and Stubbs, 2013). Care needs to be taken that the task is actually well-defined enough to pose an unambiguous challenge to capable language users.[3]

Validating a dataset is a less formalised process. It comprises arguing that the dataset indeed exemplifies the task intension well. For example, pairs only of images of giraffes and sentences describing them would arguably not exemplify the general task of *image description* very well (even if the descriptions are accurate), while perhaps exemplifying the task of *giraffe image description*.

Another way to evaluate a dataset is by trying to model it. If a model can "solve" the dataset even when deprived of information that for theoretical or pre-theoretical reasons is seen to be crucial, the dataset can be considered an unsatisfactory exemplification of the task. E.g., in a *visual (polar) question answering* setting (Antol et al., 2015), if in a dataset all and only the expressions that mention giraffes are true, a model could seize on this fact and perform well without needing the images, which would be evidence that the dataset is deficient relative to the task description.[4]

3.2 Relation Cognitive Capability / Task

While the dataset forms the visible surface of the task, it is the task itself that needs to provide value. We can categorise tasks by how they are embedded in further uses: a *product task* task is one that can be argued to have direct value to consumers (such as translation, or search); an *annotation task* is one where the task description is theoretically motivated and the output a linguistically motivated object (which may be consumed in a pipeline that itself is motivated as a product task); finally, a *benchmark task* – which is the type that concerns us here –is one which gets its value from how well it tests a particular ability (and nothing else) and how well it discriminates learners based on this ability.[5]

For a language benchmark task, the argument roughly goes as follows (even if typically only made implicitly): To be good at task T, an agent

been collecting and providing datasets for more than 20 years now (Dua and Graff, 2019).

[3] Pavlick and Kwiatkowski (2019), for example, show that the task of annotation textual entailments can lead to faultless annotator disagreements.

[4] The task of visual question answering provides an interesting example case of such a development. After Antol et al. (2015) introduced the first large scale dataset for this task, it quickly became clear that this dataset could be handled competitively by models that were deprived of visual input ("language bias", as noted e.g. by Jabri et al., 2016). This problem was then addressed by Goyal et al. (2017) with the construction of a less biased (and hence more valid) corpus for the same task.

[5] Martinez-Plumed and Hernandez-Orallo (2018), analysing AI benchmarks in general, distinguish between *difficulty* (which determines the ability level which must be reached to perform better than chance on a task) and *discrimination* (the slope of the graph plotting ability level vs. probability of correct response).

must possess a set C_T of capabilities (of representational or computational nature). If the $c \in C_T$ are capabilities that competent language users can be shown or argued to possess and make use of in using language—let's call the set of these capabilities of a competent language user C_L, so that $C_T \subseteq C_L$— then being able to model these capabilities (via modelling the task) results in progress towards the ultimate goal, which is to model competent language use. And hence, any task T that comes with an *interesting* set C_T is a good task.[6]

Under what conditions does this argument work? First of all, the assumed connection to the set of capabilities must indeed be there. We have already seen a way to challenge a claimed connection, through providing a model that can "solve" a given task (via a dataset) while not having access to information that, given our analysis of the task and interest in C_T, should be involved in the capability.[7] (Although this challenge in the first instance only targets the dataset and not the task itself.)

Secondly, following usual scientific methodology (Popper, 1934), we can rank the value of an instantiation of this argument by how precisely the capability is specified, from the trivially correct "task T involves the capability to do task T" to a statement that could be wrong, e.g. "task T involves the capability to compute the syntactic structure of a natural language sentence". Such a statement must make reference to theoretical constructs belonging to the analysis of cognitive capabilities.

Furthermore, we can rank the motivation given for a task by how explicit it is in delineating the set of capabilities it involves. For a given $c \in C_T$, is "c as required by T" fully *separable* from any other tasks involving c? Or is "c as required by T" perhaps all that there is to know about c, that is, is c *exhaustively* represented by T?

Finally, underlying the benchmarking methodology — where the benchmark is not just a measuring tool, but also a modelling target — there has to be the assumption that some sort of transferable knowledge is generated by modelling T, so that what the model (and not just the modeller!) has learned about (a sufficiently generally specified) c can be used in other tasks that involve c. (Let's call this *transferability*; which strictly speaking is a property of models, not of tasks.) More on this below.

To sum up, a benchmark task must point beyond itself and get its value from its connection to a particular facet of language, a particular capability of language users; this in turn seems to be difficult to specify without access to terms from theories of the domain, which allow us to name these capabilities.[8,9]

4 How are Language Benchmarks Used?

In the way that these tasks are set up, as single-step tasks that humans can quickly do ("describe this image", "is the elephant [in this image] sleeping?", "does sentence A follow from sentence B?"), it is tempting to see a similarity to tasks used in (human) intelligence testing (see e.g. Borsboom (2005) for an introduction). There is a crucial difference, however: Where intelligence testing works more in the way standard computing benchmarking works (subjecting the *otherwise functioning* learner to a standardised workload), in NLP, benchmarks are both the testing instrument as well as the training material.[10] The question then cannot be "to what extent does system Σ possess capability c", it has to be "to what extent can algorithm A learn c from dataset D?" — and what does that tell us?

4.1 Single-Task Models

Let's assume we have defined a task T that we are sufficiently convinced is well represented by

[6] To give some examples of informal versions of this argument, and choosing papers more or less randomly, here are some quotes:

From the paper that introduced the *visual question answering* task (Antol et al., 2015): "What makes for a compelling AI-complete task? [...] Open-ended questions require a potentially vast set of AI capabilities to answer – fine-grained recognition [...], object detection [...], activity recognition [...], knowledge based reasoning [...], and commonsense reasoning [...]."

Williams et al. (2018), on computing entailments: "The task of natural language inference (NLI) is well positioned to serve as a benchmark task for research on NLU. [...] In particular, a model must handle phenomena like lexical entailment, quantification, coreference, tense, belief, modality, and lexical and syntactic ambiguity."

[7] Such an attack challenges the claim of there being a *necessary* connection between handling T and possessing capability c. It might still very well be that humans can only perform this task if they possess capability c (and all the knowledge involved in it), because they wouldn't be able to pick up the statistical correlations that could be exploited.

[8] And one will indeed find that papers introducing such tasks make mention of terms like *syntax*, *semantics*, *compositionality*, *quantifiers*, etc.

[9] We can also note that with this focus on benchmarking normally comes a certain top-down approach, where the collected data is not investigated for how exactly the human participants went about solving their task. (But see (van Miltenburg, 2019) for a detailed study along those lines, for the task of image description.)

[10] For a recent paper also discussing the relation between AI benchmarking and intelligence testing, see (Chollet, 2019).

dataset D. We have trained a model M that performs well on this dataset. What have we learned? We have learned that a learning algorithm of the type of M can model D. Further, we have learned that the information to do task T (as exemplified in D), is contained in D, and M can pick it up.

Under what conditions can we now say that we have modelled T, rather than just D? If we have convinced ourselves that D represents T faithfully, then we might be willing to make this leap, and with it, claim that we have modelled C_T. We can get further support by collecting more data D' that also exemplifies T, but perhaps operationalises it differently. The prediction should be at least that the learning algorithm can also learn to model D'; but more significantly, we'd also want the model M learned from D to perform well on D'. Similarly, if we have another task T' of which we think that it involves similar capabilities, we should expect it to be amenable to being modelled with a learning algorithm of similar type to M.

What do we learn from a model M' (introducing architectural innovation κ over M) performing better on T (via D)? We can take this as indication that κ may be what is responsible for increasing performance, and hence what is leading to a more adequate model of C_T.

4.2 Multi-Task Models

With the advent of pre-training in NLP (Peters et al., 2018; Devlin et al., 2018), where a model is trained on (a typically large amount of) data under a specific task-regime (typically language modelling, i.e. the task of predicting the next word in a running text) and then becomes part of the model for a target task, it has become common to test on a collection of tasks (Wang et al., 2019b,a). What do we learn from such a setup? In our Figure 1, if we find a task on which we can pre-train a model M_P that becomes a part of models M and M', and which makes them more powerful than models that do not have access to the pre-trained model, then we can infer that whatever M_P models is a shared part of M and M' as well (and hence *involves* the hypothesised joint capability C'). This then provides an instrument to study the tasks: if the pre-trained model works well on some but not all, there must be something that those groups have in common. To make this intelligible, however, recourse to theoretical terms must again be taken. (E.g., assuming that these tasks involve the use of certain types of representation, or certain actions over representations.)

5 But Are We Making Progress?

Within the logic of this methodology, we are clearly making enormous progress at two links in the chain illustrated in Figure 1: For many of the established tasks, models have been and continue to be proposed that perform better, according to the metrics defined for the tasks. In addition, for many of the tasks, better datasets have been collected, avoiding exploitable biases. Where there is less activity is in systematically studying the implications of success at one task for success at others. The presentation above was largely idealised (or normative): In reality, there is very little explicitness about the assumed connection between tasks and capabilities, and no theory of how (or whether) language competence decomposes into capabilities that could be learned separately and then be assembled into a whole, and there is very little explicit knowledge about the vertical links in the Figure, from one task / model to the next.[11]

6 Conclusions

In this short paper, I have discussed the methodology of using *language tasks* to drive research on models of language competence. I have argued that the success of this approach hinges on how well progress on one task can be translated into progress on other tasks. While some steps have been taken in this direction, current work still appears to mostly focus on isolated tasks (or groups of tasks). Overcoming this, in my opinion, will require more explicit considerations about how tasks and capabilities are connected, and how the set of capabilities is structured—to ensure movement is not only uphill, but rather up the *right* hill (Bender and Koller, 2020), and it indeed is a single hill. For this, a (re-)connection with the fields that study the composition of language competence—linguistics and cognitive and developemental psychology—seems advisable (if only to disagree *explicitly*). As a positive proposal, I suggest that a focus should be put on assembling a *curriculum* of tasks, organised in a complexity and inclusion hierarchy, and that the benchmarking target should be the developmental trajectory on this. Working this out in detail I must leave for future work.

[11] In the neighbouring field of Computer Vision, there recently have been attempts to "disentangle task transfer learning" (Zamir et al., 2018).

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*.

Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistic*, 34(4):555–596.

Emily M Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2, pages 5185–5198.

Denny Borsboom. 2005. *Measuring the Mind: Conceptual Issues in Contemporary Psychometrics*. Cambridge University Press, Cambridge, UK.

Andrew Butterfield, Gerard Ekembe Ngondi, and Anne Kerr, editors. 2016. *A Dictionary of Computer Science*, 7th edition. Oxford University Press, Oxford, UK.

François Chollet. 2019. On the Measure of Intelligence. *arXiv e-prints*, page arXiv:1911.01547.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dheeru Dua and Casey Graff. 2019. UCI machine learning repository.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR 2017*.

Allan Jabri, Armand Joulin, and Laurens van der Maaten. 2016. Revisiting Visual Question Answering Baselines. In *European Conference on Computer Vision (ECCV)*.

Fernando Martinez-Plumed and José Hernandez-Orallo. 2018. Dual indicators to analyse ai benchmarks: Difficulty, discrimination, ability and generality. *IEEE Transactions on Games*, pages 1–1.

Emiel van Miltenburg. 2019. *Pragmatic factors in (automatic) image description*. Ph.D. thesis, Vrije Universiteit Amsterdam.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent Disagreements in Human Textual Inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Karl Popper. 1934. *Logik der Forschung*. Mohr Siebeck, Tübingen, Germany.

James Pustejovsky and Amber Stubbs. 2013. *Natural Language Annotation for Machine Learning*. O'Reilly, Sebastopol, CA, USA.

David Schlangen. 2019. Language tasks and language games: On methodology in current natural language processing research. *CoRR*, abs/1908.10747.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *NeurIPS*, July, pages 1–30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *ICLR 2019*, pages 1–20.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling Task Transfer Learning. In *CVPR 2018*.

X-FACT: A New Benchmark Dataset for Multilingual Fact Checking

Ashim Gupta, Vivek Srikumar
School of Computing,
University of Utah
{ashim,svivek}@cs.utah.edu

Abstract

In this work, we introduce X-FACT: the largest publicly available multilingual dataset for factual verification of naturally existing real-world claims. The dataset contains short statements in 25 languages and is labeled for veracity by expert fact-checkers. The dataset includes a multilingual evaluation benchmark that measures both out-of-domain generalization, and zero-shot capabilities of the multilingual models. Using state-of-the-art multilingual transformer-based models, we develop several automated fact-checking models that, along with textual claims, make use of additional metadata and evidence from news stories retrieved using a search engine. Empirically, our best model attains an F-score of around 40%, suggesting that our dataset is a challenging benchmark for evaluation of multilingual fact-checking models.

1 Introduction

Curbing the spread of fake news and misinformation on the web has become an important societal challenge. Several fact-checking initiatives, such as PolitiFact,[1] expend a significant amount of manual labor to investigate and determine the truthfulness of viral statements made by public figures, organizations, and social media users. Of course, since this process is time-consuming, often, a large number of falsified statements go unchecked.

With the aim of assisting fact-checkers, researchers in NLP have sought to develop computational approaches to fact-checking (Vlachos and Riedel, 2014; Wang, 2017; Pérez-Rosas et al., 2018). Many such works use the FEVER dataset, which contains claims extracted from Wikipedia documents (Thorne et al., 2018). Using real-world claims, Wang (2017) introduced LIAR, a dataset with 12,836 claims from PolitiFact. Recently, Augenstein et al. (2019) introduced MultiFC, an even larger corpus of 34,918 claims collected from 26 fact-checking websites.

Although misinformation transcends countries and languages (Bradshaw and Howard, 2019; Islam et al., 2020), much of the recent work focuses on claims and statements made in English. Developing Automated Fact Checking (AFC) systems in other languages is much more challenging, the primary reason being the absence of a manually annotated benchmark dataset for those languages. Moreover, there are fewer fact-checkers in these languages, and as a result, a non-English monolingual dataset will inevitably be small and less effective in developing fact-checking systems. As recent research points out, a possible solution in dealing with data scarcity is to train multilingual models (Aharoni et al., 2019; Wu and Dredze, 2019; Hu et al., 2020). Indeed, this finding motivates us to construct a large multilingual resource that the research community can use to further the development of fact-checking systems in languages other than English.

Recent efforts in the construction of a multilingual dataset are limited, both in scope and in size (Shahi and Nandini, 2020; Patwa et al., 2020). For instance, FakeCovid, a dataset introduced by Shahi and Nandini (2020) contains 3066 non-English claims about COVID-19. In comparison, X-FACT contains 31,189 general domain non-English claims from 25 languages. Moreover, FakeCovid contains only two labels, namely, *False*, and *Others*. We argue that this is undesirable, as fact checking is a fine-grained classification task. Due to subtle differences in language, most claims are neither entirely true nor entirely false (Rashkin et al., 2017). In contrast, our dataset contains seven labels—we make distinctions between *true*, *mostly true*, *half-true* etc. Table 1 shows two such

[1] https://www.politifact.com/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 675–682
August 1–6, 2021. ©2021 Association for Computational Linguistics

Claim	*Muslimische Gebete sind Pflichtprogramm an katholischer Schule.* Muslim prayers are compulsory in Catholic schools.
Label	Mostly-False (*Grösstenteils Falsch*)
Claimant	Freie Welt
Language	German
Source	de.correctiv.org
Claim Date	March 16, 2018
Review Date	March 23, 2018
Claim	*Temos, hoje, a despesa de Previdência Social representando 57% do orçamento.* Today, we have Social Security expenses representing 57% of the budget.
Label	Partly-True (*Exagerado*)
Claimant	Henrique Meirelles
Language	Portuguese (Brazilian)
Source	pt.piaui.folha.uol.com.br
Claim Date	None
Review Date	May 2, 2018

Table 1: Examples from X-FACT. Original labels are shown in parenthesis along with the manually mapped labels. For reference, translations are also shown.

examples from German and Brazilian Portuguese.

In summary, our contributions are:

1. We release a multilingual fact-checking benchmark X-FACT, which includes 31,189 short statements labeled for factual correctness and covers 25 typologically diverse languages across 11 language families. X-FACT is an order of magnitude larger than any other multilingual dataset available for fact checking.

2. Apart from the standard test set, we create two additional challenge sets to evaluate fact checking systems' generalization abilities across different domains and languages.

3. We report results for several modeling approaches and find that these models underperform on all three test sets in our benchmark, suggesting the need for more sophisticated and robust modeling methods.

The X-FACT dataset, and the code for our experiments, can be obtained at https://github.com/utahnlp/x-fact.

2 The X-FACT Dataset

X-FACT is constructed from several fact-checking sources. We briefly outline this process here.

Sources of Claims. We relied on a list of non-partisan fact-checkers compiled by International Fact-Checking Network (IFCN)[2], and Duke Reporter's Lab[3]. We removed all the websites that conduct fact-checks in English and are covered by previous work(Wang, 2017; Augenstein et al., 2019). As a starting point, we first queried Google's Fact Check Explorer (GFCE)[4] for all the fact-checks done by a particular website. Then we crawled the linked article on the website and additional metadata such as claimant, URL, date of the claim. For websites not linked through GFCE, we directly crawled all the available fact-checking articles from the fact-checker's website. We left out some fact-checkers because either the claims on their websites were not well specified or the fact-checker did not use any rating scale. We performed semi-automated text processing to remove duplicate claims and examples where the label appeared in the claim itself. This resulted in data from a total of 85 fact checkers for further processing. Refer to the appendix for more details on the this process.

Filtering the Dataset. There are two major challenges in using the crawled data directly: a) the labels are in different languages, and b) each fact checker uses a different rating scale for categorization. To deal with these issues, first, we manually translated all ratings to English, followed by semi-automatic merging of labels if they were found to be synonyms. Second, in consultation with Factly,[5] an IFCN signatory, we created a rating scale compatible with most fact-checkers. Our label set contains five labels with a decreasing level of truthfulness: *True*, *Mostly-True*, *Partly-True*, *Mostly-False*, and *False*. To encompass several other cases where assigning a label is difficult due to lack of evidence or subjective interpretations, we introduced *Unverifiable* as another label. A final label *Other* was used to denote cases that do not fall under the above-specified categories. Following the process described, we reviewed each fact-checker's rating system along with some examples and manually mapped these labels to our newly designed label scheme. See table 1 for examples. In our subsequent discussions, we refer to each fact-checking website as a *source*.

[2] https://www.poynter.org/ifcn/

[3] https://reporterslab.org/fact-checking/

[4] https://toolbox.google.com/factcheck/explorer

[5] https://factly.in

Data split	# claims	# languages
Train	19079	13
Development	2535	12
In-domain (α_1)	3826	12
Out-of-domain (α_2)	2368	4
Zero-Shot (α_3)	3381	12

Table 2: Dataset details. X-FACT contains three challenge sets, namely, In-domain Test (α_1), Out-of-domain Test (α_2), Zero-Shot Test (α_3).

We found that the data from several sources was dominated by a single label ($> 80\%$). Since it is difficult to train machine learning models on highly imbalanced datasets, we removed 54 such websites. We additionally removed fact-checking websites that contained fewer than 60 examples. In total, our dataset contains 31,189 fact-checks.

A Single Test Set is Not Sufficient. Recent advances in NLP have shown that multilingual models are effective for cross-lingual transfer (Kondratyuk and Straka, 2019; Wu and Dredze, 2019; Hu et al., 2020). A multilingual fact-checking system of similar transfer capabilities will certainly be an asset, especially in languages with no or few fact-checkers. From this perspective, we seek to provide a robust evaluation benchmark that can help us understand the generalization abilities of our fact-checking systems.

With this objective, we construct three test sets, namely α_1, α_2, and α_3.[6] The first test set (α_1) is distributionally similar to the training set. The α_1 set contains fact-checks from the same languages and sources as the training set.

Second, the *out-of-domain* test set (α_2), contains claims from the same languages as the training set but are from a different source. A model that performs well on both α_1 and α_2 can be presumed to generalize across different source distributions.

Third test set is the *zero-shot* set (α_3), which seeks to measure the cross-lingual transfer abilities of fact-checking systems. The α_3 set contains claims from languages not contained in the training set. Models that overfit language-specific artifacts will underperform on α_3.

Languages. For training and development, we choose the top twelve languages based on the number of labeled examples. The average number of examples per language is 1784, with Serbian being the smallest (835). We split the data into training (75%), development (10%), and α_1 test set (15%). This leaves us with 13 languages for our zero-shot test set (α_3). The remaining set of sources form our out-of-domain test set (α_2). See table 2 for the number of claims and langauges in each of these splits.

In total, X-FACT covers the following 25 languages (shown with their ISO 639-1 code for brevity): ar, az, bn, de, es, fa, fr, gu, hi, id, it, ka, mr, no, nl, pa, pl, pt, ro, ru, si, sr, sq, ta, tr. Please refer to the appendix for more details.

3 Experiments and Results

3.1 Experimental Setting

The goal of our experiments is to study how different modeling choices address the task of multilingual fact-checking. All our experiments use mBERT, the multilingual variant of BERT (Devlin et al., 2019) and use macro F1 score as the evaluation metric.[7] We report average F1 scores and standard deviations on four runs with different random seeds.

We implement the following multilingual models as baselines for future work:

1. **Claim Only Model (Claim-Only):** We provide textual claim as the only input to the model, in effect treating the problem as a simple sentence classification problem.

2. **Attention-based Evidence Aggregator (Attn-EA):** Typically, to determine the veracity of a claim, fact-checkers first gather relevant evidence by performing a web search and then aggregate this evidence to reach their final decision. We emulate this procedure by developing an attention-based evidence aggregation model that operates on evidence documents retrieved after performing web search with the claim using Google. For each claim, we obtain the top five results and use them as evidence. Using full text from web pages is not feasible, as the mBERT model has a restricted input sequence length of 512. Following previous work (Augenstein et al., 2019), we use snippets from search results as our evidence.

[6] The names for our test sets, and the idea of having multiple test sets without corresponding training sets, is inspired by Gupta et al. (2020).

[7] Although it is possible to develop partial scoring metrics, which we leave for future work to explore.

For a given claim and a collection of n evidence documents, we first encode the claim and evidences separately using mBERT by extracting the output of the `CLS` token, denoted as: $\mathbf{c}$, $[\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}]$. We first apply dot-product attention (Luong et al., 2015) to obtain the attention weights $[\alpha_1, \alpha_2, ..., \alpha_n]$, and then compute a linear combination using these attention coefficients: $\mathbf{e} = \sum_i \alpha_i \mathbf{e_i}$. This representation is then concatenated with $\mathbf{c}$ and fed to the classification layer. In all our experiments, we fix the number of evidence documents to five.

3. **Augmenting metadata (+Meta):** We concatenate additional key-value metadata with the claim text by representing it as a sequence of the form: `Key : Value` (Chen et al., 2019). This metadata includes the `language`, `website-name`, `claimant`, `claim-date`, and `review-date`. If a certain field is not available for a claim, we represent the value by `none`.

All the models are trained in a multilingual setting, i.e., a single model is trained for all languages. We could not use monolingual models as the trained monolingual models were unstable due to the small size of data for each language.

Model	α_1	α_2	α_3
Majority	6.9(–)	10.6(–)	7.6(–)
Claim-Only	38.2(0.9)	**16.2**(0.9)	14.7(0.6)
Claim-Only + Meta	39.4(0.9)	15.4(0.8)	**16.7**(1.1)
Attn-EA (Random)	37.5(0.8)	16.3(0.5)	14.9(1.2)
Attn-EA	38.9(0.2)	15.7(0.1)	16.5(0.7)
Attn-EA + Meta	**41.9**(1.2)	15.4(1.5)	16.0(0.3)

Table 3: Average F1 scores (and standard deviations) of the models studied in this work. Models in top rows are claim-only models while those in bottom are evidence-based. Attn-EA (Random) denotes the results of the evidence-based model when it is trained with random search snippets. (+ Meta) models denote those augmented with additional metadata.

3.2 Results

The results are shown in table 3. We will discuss results by answering a series of research questions. As an indicator of label distribution, we include a majority baseline with the most frequent label of the distribution (i.e. `false`).

Does the dataset exhibit claim-only bias? Before moving to more sophisticated systems, let us first examine if the model can predict a statement's veracity by only using the textual claim. Note that this setting is similar to that of hypothesis only models for the task of Natural Language Inference (NLI) (Poliak et al., 2018). From table 3, we see that a claim-only model outperforms a majority baseline by a large margin. We can draw two inferences: a) A significant number of examples in α_1 can be labeled by just relying on the textual claim, and b) the claim-only model has learned spurious correlations from the dataset.

Do search snippets improve fact-checking? First, results from table 3 show that augmenting models with metadata is helpful. Second, using search snippets as evidence with an attention-based model along with metadata improves performance by 2.5 percentage points on the in-domain test set (α_1). To further validate that snippets indeed help the evidence-based model, we perform another experiment in which we pair each claim with random search snippets of the same language. Since there is no relevant evidence, the performance is indeed similar to the claim-only model. This again confirms our finding that the dataset exhibits some claim-only bias.

While the Attn-EA model provides some performance improvement on the in-domain test set, surprisingly, the claim-only model outperforms the evidence-based model by a small margin on α_3. This might be due to the evidence-based over-fitting the in-domain data.

How informative are the search snippets? Note that we used snippets to summarize the retrieved search results. To gauge the relevance of these snippets, we manually examine 100 examples from α_1 test set for Hindi. Our preliminary analysis reveals that only 45% of snippets provide sufficient information to classify the claim, indicating why the performance increase with the evidence-based model is small. Our same analysis suggests that for 83% of the examples, using full text of the web pages provides sufficient evidence to determine veracity of the claim. Hypothetically, this means, were the models able to ingest large documents (web pages), their performance increase could have been much more significant.

Do the models generalize across sources and languages? We observe that performance on α_2

and α_3 is worse than on α_1, not only highlighting the difficulty of these challenge sets, but also showing that models overfit both source-specific patterns (α_2) and language-specific patterns (α_3).

Importantly, these results underscore the utility of our challenge sets in assessing model generalizability as well as diagnosing overfitting.

Model	α_1	α_2	α_3
X-FACT			
Claim-Only + Meta	39.4(0.9)	**15.4**(0.8)	**16.7**(1.1)
Attn-EA + Meta	**41.9**(1.2)	**15.4**(1.5)	16.0(0.3)
X-FACT + English			
Claim-Only + Meta	37.1(2.7)	14.5(0.5)	14.4(0.3)
Attn-EA + Meta	38.0(4.5)	14.7(2.6)	14.3(1.9)

Table 4: Performance comparison when augmenting the dataset with 12,311 English claims from PolitiFact. Average F1 scores (and standard deviations) of the models are reported over four random runs.

Can we improve performance by augmenting training data with English claims? Since X-FACT does not contain any examples from English, we answer this question by augmenting the training set with 12,311 claims from the PolitiFact subset of the MultiFC (Augenstein et al., 2019). Results are shown in table 4. Interestingly, we see that augmenting the models with English data hurts model performance. A possible cause is that the augmented data mostly contains political claims, while our dataset contains general claims.

4 Conclusion

We presented X-FACT, the currently largest multilingual dataset for fact-checking. Compared to the prior work, X-FACT is an order of magnitude larger, enabling the exploration of large transformer-based multilingual approaches to fact-checking. We presented results for several multilingual modeling methods and showed that the models find this new dataset challenging. We envision our dataset as an important benchmark in development and evaluation of multilingual approaches to fact-checking.

Acknowledgments

We would like to thank members of the Utah NLP group for their valuable insights, reviewers for their helpful feedback, and the team of Factly,[8] especially Mr. Shashi Kiran Deshetti, for discussions in developing a rating scale compatible with most fact-checkers. The authors acknowledge the support of NSF grants #1801446 (SATC) and #1822877 (Cyberlearning) and an award from Verisk Inc.

[8] https://factly.in/

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *EMNLP-IJCNLP*.

Samantha Bradshaw and Philip N Howard. 2019. *The global disinformation order: 2019 global inventory of organised social media manipulation*. Project on Computational Propaganda.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. TabFact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Md Saiful Islam, Tonmoy Sarkar, Sazzad Hossain Khan, Abu-Hena Mostofa Kamal, SM Murshid Hasan, Alamgir Kabir, Dalia Yeasmin, Mohammad Ariful Islam, Kamal Ibne Amin Chowdhury, Kazi Selim Anwar, et al. 2020. Covid-19–related infodemic and its impact on public health: A global social media

analysis. *The American Journal of Tropical Medicine and Hygiene*, 103(4):1621.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Parth Patwa, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2020. Fighting an infodemic: Covid-19 fake news dataset. *arXiv preprint arXiv:2011.03327*.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937.

Gautam Kishore Shahi and Durgesh Nandini. 2020. FakeCovid – a multilingual cross-domain fact check news dataset for covid-19. In *Workshop Proceedings of the 14th International AAAI Conference on Web and Social Media*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22.

William Yang Wang. 2017. "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844.

A Details on Dataset Construction

1. As mentioned in the paper, we omit several fact-checking websites from our data. A large number of these websites are not amenable to crawling and scraping the data. For instance, AFP[9] is a prominent fact-checker for many Indo-European Romance languages, but the template on its website does not lend itself to automatic data extraction tools. We can try to access this websites using GFCE, but case many times, the ratings assigned are sentences instead of a single label.

2. Another common reason is that on a number of these websites, the claim statements are not well-specified. Take for example Faktograf[10], a website performing fact-checking in Croatian. On this website , we can neither properly extract the claim statements nor do they clearly mention the rating assigned to the articles.

3. For a small percentage of the claim statements, Google search did not yield any results. We omitted all of these claims from our training, development, and test sets. These are only a very small percentage of claims, so we remove them from all models.

Because of these reasons, a large number of websites in a number of languages could not be crawled.

There are two ways we obtain our claims, labels, and other metadata. One is the Google's Fact Check Explorer (GFCE)[11], and the other is by crawling from the respective fact-checking website. In case, the links are available on GFCE, we download other metadata by visiting the website. Also, we will release the label mapping we created along with the dataset. Appendix A provides more details on the dataset we collected.

B Reproducibility

In this section, we provide details on our hyperparameter settings along with some comments on reproducibility.

[9] https://factuel.afp.com/

[10] https://faktograf.hr/ocjena-tocnosti/

[11] https://toolbox.google.com/factcheck/explorer

Dataset	Model	RunTime
X-FACT	Claim	1.5 hr
X-FACT	Claim+Meta	1.5 hr
X-FACT	Attn-EA	2.3 hr
X-FACT	Attn-EA + Meta	2.3 hr
X-FACT + Eng	Claim+Meta	2.5 Hr
X-FACT + Eng	Attn-EA + Meta	4.1 Hr

Table 5: Average Training time of the models trained

B.1 Models and Code

As described in the main paper, we used multilingual BERT for performing our experiments. We implemented all our models in PyTorch using the transformers library (Wolf et al., 2019).

B.2 Computing Infrastructure Used

All of our experiments required access to GPU accelerators. We ran our experiments on three machines: Nvidia Tesla V100 (16 GB VRAM), Nvidia Tesla P100 (16 GB VRAM), Tesla A100 (40 GB VRAM). Our experiments for the claim-only model were run on V100, and P100 GPUs and evidence-based models required larger VRAM, so they were run on A100 GPUs.

B.3 Hyperparameters and Fine-tuning Details

1. We used the mBERT-*base* model for all of our experiments. This model has 12 layers each with hiddem size of 768 and number of attention heads equal to 12. Total number of parameters in this model is 125 million. We set all the hyper-parameters as suggested by Devlin et al. (2019), except the batch size which is fixed to 8.

2. All our models were run with four random seeds (seed $= [1, 2, 3, 4]$) and the numbers reported in paper are the means of these four runs. We fine-tuned all models for ten epochs and the model performing the best on development set across all epochs was chosen as the final model.

3. Due to constraints on the VRAM of the GPUs, we restricted the number of evidence documents to five.

Average Run times Average training times are presented in table 5.

Language	ISO 639-1 code	FactChecker	Language Family	Train	Dev	α_1	α_2	α_3
Arabic	ar	misbar.com	Afro-Asiatic	✓	✓	✓		
Bengali	bn	dailyo.in	IE: Indo-Aryan					✓
Spanish	es	chequeado.com	IE: Romance	✓	✓	✓		
Persian	fa	factnameh.com	IE: Iranian					✓
Indonesian	id	cekfakta.com	Austronesian	✓	✓	✓		
Indonesian	id	cekfakta.tempo.co	Austronesian				✓	
Italian	it	pagellapolitica.it	IE: Romance	✓	✓	✓		
Italian	it	agi.it	IE: Romance				✓	
Hindi	hi	aajtak.in	IE: Indo-Aryan	✓	✓	✓		
Hindi	hi	hindi.newschecker.in	IE: Indo-Aryan				✓	
Gujarati	gu	gujarati.newschecker.in	IE: Indo-Aryan					✓
Georgian	ka	factcheck.ge	Kartvelian	✓	✓	✓		
Marathi	mr	marathi.newschecker.in	IE: Indo-Aryan					✓
Punjabi	pa	punjabi.newschecker.in.txt	IE: Indo-Aryan					✓
Polish	pl	demagog.org.pl	IE: Slavic	✓	✓	✓		
Portuguese	pt	piaui.folha.uol.com.br	IE: Romance	✓	✓	✓		
Portuguese	pt	poligrafo.sapo.pt	IE: Romance	✓	✓	✓		
Romanian	ro	factual.ro	IE: Romance	✓	✓	✓		
Norwegian	no	faktisk.no	IE: Germanic					✓
Sinhala	si	srilanka.factcrescendo.com	IE					✓
Serbian	sr	istinomer.rs	IE: Slavic	✓	✓	✓		
Tamil	ta	youturn.in	Dravidian	✓	✓	✓		
Albanian	sq	kallxo.com	IE: Albanian					✓
Albanian	sq	faktoje.al	IE: Albanian					✓
Russian	ru	factcheck.kz	IE: Slavic					✓
Turkish	tr	dogrulukpayi.com	Turkic	✓	✓	✓		
Turkish	tr	teyit.org	Turkic				✓	
Azerbaijani	az	faktyoxla.info	Turkic					✓
Portuguese	pt	aosfatos.org	IE: Romance					✓
German	de	correctiv.org	IE: Germanic	✓	✓	✓		
Dutch	nl	nieuwscheckers.nl	IE: Germanic					✓
French	fr	fr.africacheck.org	IE: Romance					✓

Table 6: Details of the X-FACT dataset. Our dataset belongs to 25 typologically diverse languages across 11 language families. The table shows the composition of training, development, and three challenge sets. IE: denotes Indo-Aryan

nmT5 - Is parallel data still relevant for pre-training massively multilingual language models?

Mihir Kale* **Aditya Siddhant*** **Rami Al-Rfou**
Linting Xue **Noah Constant** **Melvin Johnson**
Google Research

Abstract

Recently, mT5 - a massively multilingual version of T5 - leveraged a unified text-to-text format to attain state-of-the-art results on a wide variety of multilingual NLP tasks. In this paper, we investigate the impact of incorporating parallel data into mT5 pre-training. We find that multi-tasking language modeling with objectives such as machine translation during pre-training is a straightforward way to improve performance on downstream multilingual and cross-lingual tasks. However, the gains start to diminish as the model capacity increases, suggesting that parallel data might not be as essential for larger models. At the same time, even at larger model sizes, we find that pre-training with parallel data still provides benefits in the limited labelled data regime.

1 Introduction

Recent works have shown that cross-lingual transfer learning in pre-trained multilingual models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) could be improved further by using parallel data (Conneau and Lample, 2019; Hu et al., 2020a; Ouyang et al., 2020; Luo et al., 2020). In this paper, we continue this line of work by improving the recent mT5 model (Xue et al., 2020) by leveraging parallel corpora. We experiment with several text-to-text objectives that incorporate parallel data (spanning 198 language pairs) into mT5 pre-training. Our key findings are summarized below:

- In the regime of very small fine-tuning datasets, objectives with parallel data improve results significantly.
- The gain from using parallel data decreases as we scale up the size of the pre-trained model.
- Simple objectives based on neural machine translation (NMT) perform better than the traditionally employed "translation language modeling" (TLM) objective.

*Equal Contribution. Please direct correspondence to {mihirkale, adisid}@google.com

2 Method

We focus on the mT5-Large model, which is a 24 layer encoder-decoder transformer model and has shown strong performance on a variety of cross-lingual benchmarks (Xue et al., 2020). Instead of training a new model from scratch, we start from the publicly available mT5-Large checkpoint - which has been trained for over 1 trillion tokens - and do a second stage pre-training with a mix of monolingual and parallel data.

2.1 Objectives

The mT5 - multilingual version of T5 (Raffel et al., 2020) - series of models were pre-trained on a multilingual version of the C4 corpus with a masked language modeling "span-corruption" objective (Raffel et al., 2020), where the encoder is fed a chunk of text with random spans replaced with a mask token, and the decoder must reconstruct the masked-out tokens. One of their primary distinctions is the use of a unified "text-to-text" format for all text-based NLP problems.

In keeping with the text-to-text format, we experiment with the following objectives to incorporate parallel data into pre-training:

- **TLM** - A text-to-text version of translation language modeling, proposed by Conneau and Lample (2019) and subsequently used in several prior works for encoder only pre-training. We trivially extend it to the encoder-decoder setting.
- **NMT** - Standard machine translation. The input is the source text and the target is its

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 683–691
August 1–6, 2021. ©2021 Association for Computational Linguistics

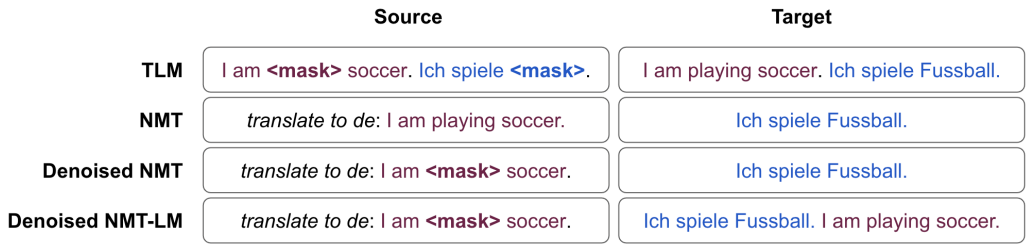


Figure 1: Example source and targets for different text-to-text style pre-training objectives incorporating parallel data. All objectives except TLM specify target language in the source sentence.

translation. A language code is prefixed to the input to inform the model of the target language (Johnson et al., 2017).

- **Denoised-NMT** - Similar to NMT, but we additionally mask spans in the source sentence. The model must now learn to implicitly perform language modeling of the source language while translating into the target language.
- **Denoised-NMT+LM** - Similar to Denoised-NMT, but instead of implicit language modeling, the model must explicitly predict the source text in addition to the translation. The target is a concatenation of the translation and source sentence, while the input is the masked source sentence.

We refer to the model trained with the standard NMT objective as nmT5.

3 Experiment Setup

Pre-training datasets For pre-training we use monolingual data from mC4 (Xue et al., 2020) and parallel data from OPUS-100 (Zhang et al., 2020). OPUS-100 is a dataset of 55M translations covering 100 languages (198 language pairs, either into or from English). The mC4 corpus consists of unlabeled web text covering 101 languages, of which 81 overlap with the OPUS-100 languages.

Fine-tuning datasets For downstream evaluation, we use the following four tasks:

- **TyDi QA** (Clark et al., 2020) - The GoldP subtask, which corresponds to extractive question answering. The input is a passage and a question, with the answer being a span from the passage.
- **MTOP** (Li et al., 2020) - Multilingual Task-Oriented Parsing. The task is one of structured prediction, where user queries must be parsed into a tree, capturing the domain, intent and slots.
- **WikiAnn NER** (Pan et al., 2019) - Named entity recognition task covering 40 languages featured in the XTREME benchmark (Hu et al., 2020b). There are 4 categories of entities - location, person, organization and miscellaneous.
- **WikiLingua** (Ladhak et al., 2020) - A recently introduced *cross-lingual* summarization dataset, where a document from an arbitrary language must be summarized in English. Since the dataset does not come with training and evaluation splits, we randomly create validation and test sets of 1000 examples each, and the rest of the data is used for training.

Dataset	Langs	Train size	Setting
TyDi QA	9	3.7K	zero-shot
MTOP	6	22K	zero-shot
WikiAnn NER	40	20K	zero-shot
WikiLingua	18	660K	multilingual

Table 1: Statistics of datasets used in the paper.

Table 1 lists further details of each dataset. Following Xue et al. (2020), all tasks are cast into the text-to-text format. The evaluation for TyDi QA, MTOP and NER is done in the zero-shot setting, where the model is trained on the English data and evaluated on all languages. Since zero-shot cross-lingual language generation is much harder, for WikiLingua we train the model in a multilingual setting, using available training data for all languages.

Hyperparameters Pre-training is done with a batch size of 1M tokens and fine-tuning with 131,072 tokens, with a constant learning rate of

Model (Metric)	TyDi QA (F1/EM)	MTOP (EM)	NER (F1)	WikiLingua (ROUGE-L)	Avg.
mT5	66.3 / 49.8	43.7	58.4	25.2	46.3
+MLM (additional 100K steps)	71.3 / 55.6	48.6	59.9	26.1	49.5
+MLM+TLM	71.1 / 54.6	48.6	61.4	26.1	49.7
+MLM+NMT	75.1 / 60.1	57.7	61.4	27.4	53.5
+MLM+denoised NMT	75.3 / 60.2	56.5	61.5	27.4	53.3
+MLM+denoised NMT-LM	75.0 / 59.4	56.0	62.4	26.9	53.1

Table 2: Results are averaged across all the languages in each dataset. We report F1/EM for QA, exact match accuracy (EM) for structured prediction, ROUGE-L (Lin, 2004) for summarization and F1 for NER. Each score is the median over five runs. The final columns lists the average of all the scores. Refer to Appendix A for scores on individual languages.

0.001. Starting from publicly available mT5-Large checkpoints, we further pre-train for 100K steps with a mix of monolingual and parallel objectives. The parallel data is mixed into monolingual data at a 10% ratio, which amounts to roughly 4 passes over the OPUS-100 corpus. Examples from each language pair are sampled using the same language sampling distribution as Xue et al. (2020), with alpha=0.3. For downstream tasks, we fine-tune for 10K steps for TyDiQA, MTOP, NER and 25K for WikiLingua, since it is a much larger dataset. Checkpoint selection is done based on the validation set.

Baselines Our first baseline is the publicly available mT5-Large model (1.3 billion parameters). For a fair comparison, we also experiment with an mT5 model further pre-trained for 100k steps with *only* monolingual data from mC4 (see row 2: mT5+MLM in Table 2). This lets us assess whether improvements stem from using parallel data or just pre-training for longer.

4 Results

We report results in table 2. Overall, adding parallel data through neural machine translation objectives improves scores for all 4 tasks, with the NMT objective performing the best.

Simply pre-training mT5 for longer with just monolingual data (MLM) leads to improved scores for all tasks. The TLM objective is not be able to effectively leverage the parallel data and performs on par with MLM. On the other hand, our three NMT-based objectives show gains over MLM across all tasks. Among these, NMT and Denoised-NMT are the best and perform similarly, while Denoised-NMT+LM fares slightly worse. Averaged across all tasks, NMT and Denoised-NMT outperform MLM by 4 points.

4.1 Model size

Xue et al. (2020) find that cross-lingual performance of language models increases monotonically with model size. To study the impact of model capacity, we also experiment with larger model sizes. Even at the XL size (3.7B params, 3× larger than mT5-Large), we observe gains for all tasks with nmT5 (Table 3). However, the magnitude of the gains is largely diminished, hinting that the need for parallel data reduces as model capacity increases. This finding is particularly promising for low-resource languages, where it is difficult to obtain high-quality parallel data.

At the same time, nmT5-Large substantially reduces the performance gap between mT5-Large and mT5-XL, covering 70% of the headroom. Since bigger models are expensive to train and even more expensive to deploy, this opens up avenues for effectively using parallel data to improve performance of smaller language models. Turc et al. (2019) found that pre-training student models before model distillation is helpful, and using parallel data to improve student pre-training is another interesting avenue of future work.

Model	TyDi QA	MTOP	NER	WikiLingua	Avg.
mT5-Large	66.3 / 49.8	43.7	58.4	25.2	46.3
nmT5-Large	75.1 / 60.1	57.7	61.4	27.4	53.5
Δ	8.8 / 10.3	14.0	3.0	2.2	7.2
mT5-XL	77.8 / 61.8	63.4	65.5	27.9	56.7
nmT5-XL	78.4 / 63.3	64.9	66.2	28.4	57.6
Δ	0.6 / 1.5	1.5	0.7	0.5	0.9

Table 3: Impact of model size on nmT5's performance.

Model	Few-Shot (100)	Low (3.7K)	High (80K)
mT5-Large	33.1 / 23.6	66.3 / 49.8	78.1 / 64.8
nmT5-Large	48.8 / 37.1	75.1 / 60.1	78.2 / 65.5
Δ	15.7 / 13.5	8.8 / 10.3	0.1 / 0.7
mT5-XL	45.0 / 31.7	77.8 / 61.8	78.7 / 65.8
nmT5-XL	57.2 / 44.4	78.4 / 63.3	79.7 / 67.0
Δ	12.2 / 12.7	0.6 / 1.5	1.0 / 1.2

Table 4: Performance on the TyDi QA eval set when fine-tuned in the *few-shot* (100 examples from TyDi QA English), *low* (full TyDi QA English with 3.7K examples) and *high* data regime (SQuAD English with 80K examples).

4.2 Limited labeled data

The TyDi QA dataset has only 3.7K English training examples. To study the impact of the size of fine-tuning data, we run experiments in two additional settings: a *few-shot* regime and a *high data* regime. Few-shot uses just 100 randomly sampled training examples, while for the latter we use the much larger SQuAD corpus (Rajpurkar et al., 2016), which consists of 80k examples.

When fine-tuned with SQuAD, nmT5 performs slightly better than mT5 for both Large and XL model sizes. However, in the few-shot setting, nmT5-Large improves over mT5-Large by 15 points. Even at the XL size, nmT5 is over 10 points higher than mT5. nmT5-Large even outperforms the much larger mT5-XL. Our experiments suggest that pre-training with parallel data is particularly useful in the limited labelled data setting.

4.3 Mixing ratio

So far, we have mixed parallel data into monolingual data at a 10% ratio. To assess how the mixing ratio impacts performance, we compare results with a 50% mix. With the 50% mix, average performance is slightly lower, validating our initial choice.

Mix	TyDi QA	MTOP	NER	WikiLingua	Avg.
10%	75.1 / 60.1	57.7	61.4	27.4	53.5
50%	76.5 / 60.1	53.9	62.0	26.5	52.7

Table 5: Impact of mixing ratio on nmT5.

4.4 Performance on unseen languages

We also test downstream performance on languages previously unseen by the models. We randomly pick 30 languages from the WikiAnn NER dataset that are not covered in either mC4 [1] or OPUS, and hence none of our models have seen them during pre-training. Table 6 shows nmT5 outperforms mT5 on this subset of languages as well, indicating that the representations of the nmT5 model are better suited for cross-lingual transfer.

Model	ckb	hsb	xmf	"Avg."
mT5-Large	66.5	64.8	58.4	54.9
nmT5-Large	72.2	69.8	62.2	57.4
Δ	5.7	5.0	3.8	2.5

Table 6: Performance on three randomly picked unseen languages. "Avg." is calculated by averaging performance across 30 unseen languages.

5 Related Work

Pre-trained multilingual models such as mBERT and XLM-R have shown to be effective at cross-lingual transfer learning (Devlin et al., 2019; Conneau et al., 2020). Subsequently, many attempts have leveraged parallel data to improve cross-lingual capability of these models. Conneau and Lample (2019) proposed translation language modeling (TLM), to encourage the model to align representations across languages. Alternating language modeling (Yang et al., 2020) and back-translation masked language modeling (Ouyang et al., 2020) used code-switched sentences and back-translation respectively to utilize parallel data. Other works using parallel data in this line of work include FILTER (Fang et al., 2020), AMBER (Hu et al., 2020a) and, MMTE (Siddhant et al., 2020). A key factor that differentiates this paper from these works is that our pre-trained models use a text-to-text architecture, having both an encoder and a decoder, while the aforementioned models only have the encoder. Other pretrained multilingual encoder-decoder models such as mT5 (Xue et al., 2020), mBART (Liu et al., 2020) and MASS (Song et al., 2019) do not make use of parallel data during pre-training.

6 Conclusion

In this work we attempted to improve mT5 pre-training by incorporating parallel data. We experimented with various text-to-text objectives and found that multi-tasking with the standard neural machine translation objective during pre-training

[1] Subject to precision of language ID models used for mC4.

leads to improved cross-lingual transfer. The improvements from parallel data are most pronounced in the limited labeled data scenario. Our experiments also indicate that smaller models, with the help of parallel data, can approach the performance of larger ones, while also suggesting that the need for parallel data is lesser as the model capacity increases.

References

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. Filter: An enhanced fusion method for cross-lingual language understanding. *arXiv preprint arXiv:2009.05166*.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2020a. Explicit alignment objectives for multilingual bidirectional encoders. *arXiv preprint arXiv:2010.07972*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020b. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*.

M. Johnson, Mike Schuster, Quoc V. Le, M. Krikun, Y. Wu, Z. Chen, Nikhil Thorat, F. Viégas, M. Wattenberg, G. S. Corrado, Macduff Hughes, and J. Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Faisal Ladhak, Esin Durmus, Claire Cardie, and K. McKeown. 2020. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization. *ArXiv*, abs/2010.03093.

Haoran Li, A. Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *ArXiv*, abs/2008.09335.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and Luo Si. 2020. Veco: Variable encoder-decoder pre-training for cross-lingual understanding and generation. *arXiv preprint arXiv:2010.16046*.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*.

Xiaoman Pan, Thamme Gowda, Heng Ji, Jonathan May, and Scott Miller. 2019. Cross-lingual joint entity and word embedding to improve entity linking and parallel sentence mining. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 56–66, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Ari, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. 2020. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv: Computation and Language*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Jian Yang, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

A Per-Language Results on All Tasks

	en	ar	bn	fi	id
mt5	75.0 / 63.0	68.9 / 51.4	54.5 / 37.2	70.4 / 54.6	74.3 / 57.0
+MLM	78.5 / 68.2	76.1 / 59.9	59.0 / 40.7	73.5 / 61.0	76.7 / 60.0
+MLM+TLM	77.3 / 67.0	75.7 / 57.2	61.7 / 39.8	73.3 / 59.0	77.0 / 60.0
+MLM+NMT	78.4 / 69.3	78.9 / 63.1	74.0 / 54.9	77.0 / 64.8	79.9 / 64.8
+MLM+denoised NMT	78.7 / 68.6	79.8 / 64.7	72.6 / 53.1	77.2 / 64.2	79.8 / 67.6
+MLM+denoised NMT-LM	78.2 / 68.2	78.8 / 62.3	69.1 / 49.6	78.2 / 65.7	79.6 / 64.8
	ko	ru	sw	te	avg
mt5	57.4 / 47.5	61.5 / 37.1	69.7 / 52.5	65.5 / 48.0	66.3 / 49.8
+MLM	64.4 / 55.4	68.6 / 48.9	74.2 / 57.7	71.1 / 48.6	71.3 / 55.6
+MLM+TLM	66.5 / 55.8	67.8 / 48.0	73.9 / 57.1	66.5 / 47.5	71.1 / 54.6
+MLM+NMT	64.9 / 56.2	72.1 / 51.8	77.2 / 63.1	73.3 / 53.1	75.1 / 60.1
+MLM+denoised NMT	67.9 / 58.7	71.9 / 51.5	75.7 / 59.7	74.3 / 53.5	75.3 / 60.2
+MLM+denoised NMT-LM	67.8 / 59.4	72.7 / 51.1	76.0 / 59.9	74.4 / 54.0	75.0 / 59.4

Table 7: TyDi QA GoldP results (F1/EM) for each language.

	en	de	es	fr	hi	th	avg
mt5	83.5	41.2	45.4	43.3	21.3	27.5	43.7
+MLM	83.3	44.5	46.3	51.8	31.9	34.0	48.6
+MLM+TLM	85.0	42.4	47.5	49.6	31.8	35.2	48.6
+MLM+NMT	86.1	55.1	59.0	61.7	42.2	42.1	57.7
+MLM+denoised NMT	85.8	51.6	55.2	59.5	42.7	43.9	56.5
+MLM+denoised NMT-LM	85.9	51.9	55.0	57.0	44.1	41.9	56.0

Table 8: MTOP results (EM) for each language.

	en	af	ar	bg	bn	de	el	es	et	eu	fa	fi	fr	he
mt5	80.5	64.5	47.7	57.2	66.5	67.0	63.9	62.0	59.0	45.5	41.4	56.9	76.7	45.1
+MLM	81.4	65.1	50.2	55.2	69.3	68.6	66.9	70.5	62.8	46.6	44.9	58.9	76.6	46.4
+MLM+TLM	82.4	65.6	48.8	67.2	72.2	70.1	70.8	72.6	61.2	47.5	47.1	61.4	78.7	48.0
+MLM+NMT	82.2	64.2	56.7	61.0	69.1	70.5	64.6	66.3	66.2	49.3	48.9	60.6	78.4	46.2
+MLM+denoised NMT	82.5	65.7	50.3	63.6	69.6	70.7	68.6	73.7	64.9	48.6	44.3	63.3	77.7	45.5
+MLM+denoised NMT-LM	82.9	66.1	49.5	67.7	74.5	71.1	71.3	74.2	67.1	49.9	44.8	63.2	80.2	49.6
	hi	hu	id	it	ja	jv	ka	kk	ko	ml	mr	ms	my	nl
mt5	66.8	57.7	44.9	75.4	36.0	46.0	53.0	22.5	29.5	44.8	38.6	65.5	27.0	77.3
+MLM	66.5	61.4	46.2	76.4	35.8	49.0	53.6	23.7	31.4	46.0	39.3	67.4	33.0	78.5
+MLM+TLM	69.6	61.9	47.2	76.7	37.3	51.0	59.4	29.3	30.7	48.2	42.1	70.2	29.0	80.4
+MLM+NMT	69.8	61.7	46.1	77.3	34.5	53.0	55.2	27.0	31.4	43.0	46.7	69.0	27.0	78.9
+MLM+denoised NMT	65.8	63.0	46.6	77.6	37.0	54.0	58.3	26.4	29.8	44.8	42.1	64.3	30.0	80.2
+MLM+denoised NMT-LM	67.7	64.4	48.1	77.9	39.2	49.0	59.4	30.0	31.4	47.4	36.4	71.0	34.0	80.2
	pt	ru	sw	ta	te	th	tl	tr	ur	vi	yo	zh	avg	
mt5	73.1	48.4	66.8	39.9	37.9	8.5	77.8	57.6	45.1	76.4	58.0	41.8	58.4	
+MLM	75.5	47.3	64.5	40.5	38.0	9.2	76.9	56.5	51.7	76.9	59.0	41.8	59.9	
+MLM+TLM	76.3	58.8	66.3	40.2	41.2	8.8	76.9	62.0	43.0	79.6	56.0	43.5	61.4	
+MLM+NMT	75.5	56.0	65.8	40.3	41.6	8.0	78.7	60.3	57.0	79.8	63.0	41.0	61.4	
+MLM+denoised NMT	75.5	58.9	66.2	40.4	40.4	7.9	78.7	60.5	50.0	80.3	64.0	41.4	61.5	
+MLM+denoised NMT-LM	78.6	60.9	65.6	40.6	40.9	9.1	77.0	63.1	53.5	79.8	60.0	45.5	62.4	

Table 9: WikiAnn NER results (F1) for each language.

	en	ar	cs	de	es	fr	hi	id	it	ja
mt5	29.2	23.2	22.4	25.0	25.3	24.6	25.2	25.3	24.1	26.2
+MLM	30.0	24.0	22.9	26.0	26.6	25.5	26.1	25.8	24.9	27.8
+MLM+TLM	30.0	24.4	23.1	25.6	26.3	25.6	26.4	25.8	25.1	27.6
+MLM+NMT	31.5	25.7	24.0	27.0	27.5	26.4	27.7	27.0	25.8	29.5
+MLM+denoised NMT	31.3	25.7	24.7	27.3	27.5	26.8	27.8	27.2	25.8	29.2
+MLM+denoised NMT-LM	30.8	25.0	23.7	26.5	27.1	26.3	27.3	26.7	25.6	28.7
	ko	nl	pt	ru	th	tr	vi	zh	avg	
mt5	23.8	25.7	24.6	23.9	25.3	30.9	22.9	25.8	25.2	
+MLM	25.2	26.5	25.3	24.6	27.1	31.1	23.2	27.1	26.1	
+MLM+TLM	24.7	26.6	25.2	24.4	26.5	31.3	23.3	27.0	26.1	
+MLM+NMT	26.7	27.7	26.3	25.9	28.6	34.1	23.9	28.1	27.4	
+MLM+denoised NMT	26.6	28.0	25.9	25.8	28.3	33.4	24.3	28.4	27.4	
+MLM+denoised NMT-LM	25.9	27.4	25.6	24.9	27.3	33.1	23.8	27.8	26.9	

Table 10: Wikilingua results (Rouge-L) for each language.

	ace	arz	ast	ba	ce	ckb	csb	eml	fur	gan	gn
mt5-Large	44.8	50.8	83.3	38.1	21.7	66.5	56.7	39.8	64.2	42.1	48.2
nmt5-Large	46.7	53.6	84.8	43.7	28.3	72.2	58.1	41.9	65.6	41.2	51.0
	hsb	ia	jbo	lij	lmo	min	nap	nov	pdc	pms	pnb
mt5-Large	64.8	63.2	42.1	46.3	69.8	39.1	62.2	62.1	48.1	81.5	61.1
nmt5-Large	69.8	62.4	43.6	43.0	72.0	45.5	61.7	66.7	51.2	83.5	55.4
	rm	sa	tl	qu	vec	vep	vls	xmf	avg		
mt5-Large	64.1	17.4	78.6	27.5	66.9	63.6	74.4	58.4	54.9		
nmt5-Large	67.6	23.0	79.4	35.6	66.7	68.0	77.5	62.2	57.4		

Table 11: WikiAnn NER results on unseen languages. Refer to section 4.4

Question Generation for Adaptive Education

Megha Srivastava
Stanford University
megha@cs.stanford.edu

Noah Goodman
Stanford University
ngoodman@stanford.edu

Abstract

Intelligent and adaptive online education systems aim to make high-quality education available for a diverse range of students. However, existing systems usually depend on a pool of hand-made questions, limiting how fine-grained and open-ended they can be in adapting to individual students. We explore targeted question generation as a controllable sequence generation task. We first show how to fine-tune pre-trained language models for deep knowledge tracing (LM-KT). This model accurately predicts the probability of a student answering a question correctly, and generalizes to questions not seen in training. We then use LM-KT to specify the objective and data for training a model to generate questions conditioned on the student and target difficulty. Our results show we succeed at generating novel, well-calibrated language translation questions for second language learners from a real online education platform.

1 Introduction

Online education platforms can increase the accessibility of educational resources around the world. However, achieving equitable outcomes across diverse learning needs benefits from systems that are adaptive and individualized to each student (Doroudi and Brunskill, 2019). Traditionally, adaptive education methods involve planning over a pool of pre-made questions (Atkinson, 1972; Hunziker et al., 2018). These are naturally limited by the diversity and coverage of the pool, as well as the scaling capacity of curriculum planning algorithms. Recent approaches, such as procedural generation for personalized programming games (Valls-Vargas et al., 2017), are limited to well-specified small domains. We address these limitations by leveraging recent success in deep generative models, in particular language models (LMs).

Many educational activities involve sequential data, such as language translation, reading comprehension, algebra, and deductive logic. Meanwhile, pre-trained LMs can effectively handle sequences from a wide range of modalities (Madani et al., 2020; Polu and Sutskever, 2020). In this work, we focus on natural language sequences, where recent progress in language modeling has shown great success at capturing abstract properties of language (Hewitt and Manning, 2019; Liu et al., 2019). Specifically, we show how pre-trained LMs can be easily leveraged to adaptively generate questions for a given student and target difficulty in a *reverse translation task*, using difficulty at answering questions as a proxy for more complex future learning objectives.

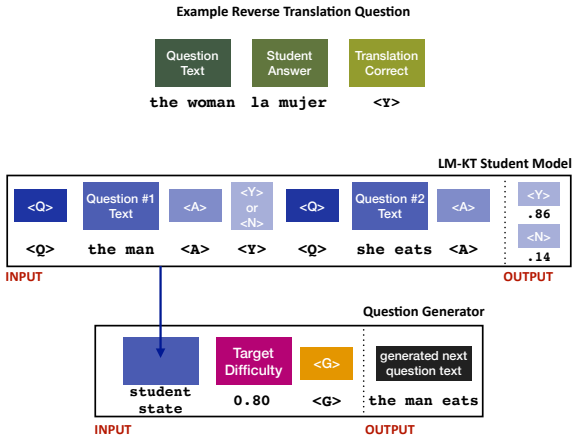


Figure 1: Example input and outputs for our LM-based knowledge tracing model (middle) and question generation model (bottom) for an online reverse language translation task (top). A question in this task consists of a target phrase for the student, in this case a Spanish learner, to translate (e.g. "the woman").

We introduce an LM-based knowledge tracing model (LM-KT) to predict students' difficulty on novel questions (e.g. target phrases to translate). We show that LM-KT is well-calibrated, allowing us to pose the learning problem for the question generator: given a student state, generate a question that will achieve a target difficulty, according

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 692–701
August 1–6, 2021. ©2021 Association for Computational Linguistics

to LM-KT. We evaluate both LM-KT and question generation models on real users and responses from Duolingo[1], a popular online second-language learning platform.

2 Background & Related Works

There exists a rich body of work on precisely modeling student "ability" and learning. For example, Item Response Theory (IRT) seeks to model individual student ability based on their responses to different questions, creating a strong factorization between students and test items (Lord, 1980; Hambelton and Jodoin, 2003). Meanwhile, Computer Adaptive Testing (CAT) techniques are used to determine a fixed student ability as quickly as possible by selecting test items based on information utility (Weiss and Kingsbury, 1984; Thissen and Mislevy, 2000; Settles et al., 2020). However, these methods, which have been used to develop efficient standardized tests, do not necessarily optimize a student's *learning* experience (Mu et al., 2018). We instead focus on tracking each student's evolving knowledge, choosing questions to target difficulty.

Knowledge Tracing (KT) seeks to model a student's knowledge state from their answer history in order to help individualize exercise sequences (Corbett and Anderson, 1995). This draws inspiration from traditional education curriculum practices, such as distributed spacing of vocabulary (Bloom and Shuell, 1981) and mixed review in mathematics (Rohrer, 2009). To address simplifying assumptions in earlier KT approaches, such as discrete knowledge representations, Piech et al. (2015) introduced Deep Knowledge Tracing (DKT), which uses RNNs to enable more complex knowledge representations for students. Recently, SAINT+ (Shin et al., 2020) showed state-of-the-art performance on the popular EdNet KT task using a Transformer model to capture temporal information across activities, motivating our use of Transformer LMs.

Controllable Text Generation aims to steer LMs towards desired attributes. Examples include using reinforcement learning to control quality metrics (Ranzato et al., 2016), adjusting sampling weights to control for poetry style (Ghazvininejad et al., 2017), and learning to condition on valence or domain-specific codes (Keskar et al., 2019; Peng et al., 2018). To the best of our knowledge, we are the first to use controllable generation in an education context with real student interaction data.

[1] http://duolingo.com

3 Method

Given any autoregressive language model (e.g. GPT-2 (Radford et al., 2019), we can fine-tune a **LM-KT model** ($p_{\theta_{KT}}$) to predict whether an individual student will correctly answer the next question. If this model has well-calibrated uncertainty, we can use its predicted probability of a correct answer as a proxy for the difficulty of a question to a student. We then train a **question generation model** ($p_{\theta_{QG}}$) to generate a new question conditioned on a student and desired target difficulty.

Question Representation Unlike standard DKT, which treats questions as IDs or simple handcrafted features, we represent questions fully in text (e.g. "she eats" in Figure 1). This is a key contribution of our work, required by our eventual goal of *generating* questions in text, and allows the model to leverage similarity across linguistic features. We thus represent a question q as a sequence of words, with prefix and suffix tokens:

$$q_i = \texttt{<Q>}\ w_1^i\ w_2^i\ w_3^i\ \ldots\ w_n^i\ \texttt{<A>}$$

Student State We represent a student as a temporally-evolving sequence of questions and their responses. As in much previous KT work, we represent the student response as simply correct/incorrect, with special tokens <Y> and <N>. A student's current state is thus represented as a sequence of all past question and response pairs:

$$s_j = q_1^j\ a_1^j\ q_2^j\ a_2^j\ \ldots\ q_m^j\ a_m^j\ , a_i \in \{\texttt{<Y>},\texttt{<N>}\}$$

LM-KT Given the sequential nature of student learning over time, we can easily frame knowledge tracing as an autoregressive language modeling task. Given a dataset D of students s_1, s_2, ..., $s_{|D|}$, we employ the standard training objective of finding the parameters θ_{KT} that minimizes

$$\mathscr{L}_{KT} = -\sum_{i=1}^{|D|}\sum_{t=1}^{|\mathbf{x}^{(i)}|} \log p_{\theta_{KT}}(x_t^{(i)}|x_{<t}^{(i)}) \quad (1)$$

where $\mathbf{x}^{(j)} = (x_1^{(j)}, ..., x_{|\mathbf{x}|}^{(j)})$ is the entire sequence tokens corresponding to student s_j, consisting of all their past questions and answers. Using the softmax output of the LM-KT model ($p_{\theta_{KT}}$), we estimate a student's (inverse) difficulty in answering a specific question as $d_{qs} = p_{\theta_{KT}}(\texttt{<Y>}|s, q)$. We find that $p_{\theta_{KT}}$ is well-calibrated (Section 4.2), yielding a good proxy for the true question difficulty.

Question Generation We frame question generation as finetuning a *new* autoregressive LM. Given random samples of students and questions from a held-out set not used to train LM-KT, we can construct a new dataset D' consisting of s_i d_i<G> q_i sequences, where <G> is a special generation token and $d_i = p_{\theta_{KT}}(\texttt{<Y>}|s_i, q_i)$ is the continuous difficulty value assigned by LM-KT. We learn a linear layer to map the continuous input difficulty into a *difficulty control vector* c_d of dimension matching the LM word-embeddings, which we append to the token embeddings. Unlike LM-KT, we train our question generation model $p_{\theta_{QG}}$ to minimize the loss only on the question text, which only appears *after* the<G> token. If t_g is the token index of <G>, then our modified loss is:

$$\mathscr{L}_{QG} = -\sum_{i=1}^{|D'|}\sum_{t=t_g+1}^{|\mathbf{x}^{(i)}|} \log p_{\theta_{QG}}(x_t^{(i)}|x_{<t}^{(i)}) \quad (2)$$

where sequence $\mathbf{x}^{(j)}$ contains the full s_j d_j<G>q_j sequence. At test time, we generate tokens $w_1 ... w_n$ conditioned on the s_j d_j <G> prefix.

4 Experiments

Our method generalizes to any education activity that can be represented with text sequences. Due to the availability of real student learning data, we focus on a *reverse language translation* task, where a student translates phrases from their native language (e.g. English, "she eats") to the second language they are learning (e.g. Spanish, "ella come").

4.1 Experimental Details

We use the 2018 Duolingo Shared Task on Second Language Acquisition Modeling (Settles et al., 2018) dataset, which contains questions and responses for Duolingo users over the first 30 days of learning a second language. While the original task's goal was to identify *token*-level mistakes, we collapse these errors into binary (correct / incorrect) per-question labels. We use the provided train/dev/test splits for users learning Spanish and French. We create separate held-out sets from the test set to evaluate the LM-KT and question generation models. For both models, we finetune separate GPT-2 (Radford et al., 2019) models. While we sample from a held-out set of student states and questions to train the question generation model, in principle questions can come from any source text domain. Further experiment details are in the Appendix, and source code can be found at: https://github.com/meghabyte/acl2021-education.

Model *(Spanish)*	**AUC (seen)**	**AUC (unseen)**
LM-KT	**0.75 ±.0001**	**0.76 ±.001**
Standard DKT	0.72 ±.0001	0.70 ±.001
Question Only	0.67 ±.0001	0.58 ±.002
Model *(French)*	**AUC (seen)**	**AUC (unseen)**
LM-KT	**0.73 ±.0002**	**0.71 ±.002**
Standard DKT	0.70 ±.0001	0.65 ±.002
Question Only	0.65 ±.0002	0.62 ±.001

Table 1: LM-KT improves AUC for both questions in the Duolingo test set that were seen during training (for other students) *and* novel questions, over Standard DKT with Question IDs and question-only baselines. Errors are 95% CIs.

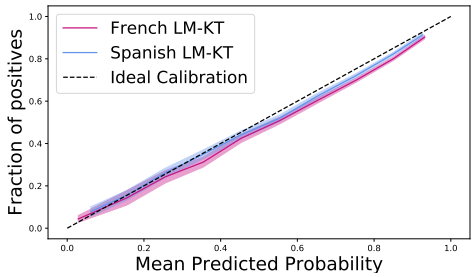


Figure 2: Both LM-KT models are well calibrated, but the French model is slightly more overconfident. Filled area shows bootstrap (n=1000) standard deviation.

4.2 Results: Student Modeling

We evaluate LM-KT two ways: first, its ability to predict if an individual student will answer a novel question correctly on a held-out test set of real Duolingo student responses. Second, how well-calibrated these predictions are, which is crucial to our later use of LM-KT for question generation.

Table 1 compares AUC-ROC on a held-out test set for our LM-KT model with standard DKT, which uses question IDs instead of text, and a baseline that ignores the student state, only using the question text representation. This question only baseline would perform well if the Duolingo dataset largely consisted of universally "easy" and "difficult" questions, independent of individual student. Our results show that incorporating the student state is crucial for accurately predicting Duolingo user responses, and including question text also leads to a significant improvement. LM-KT outperforms Standard DKT especially on novel questions—a necessary generalization ability for generation.

Finally, we measure the calibration of our LM-KT models for both Spanish and French (from En-

glish) learners, which is the crucial property for our downstream generation task. We bin our test data by predicted question difficulty, and plot the fraction of true correct answers in each bin. Figure 2 shows that LM-KT is well-calibrated, for both Spanish and French, meaning the predicted difficulty matches the empirically observed proportion of correct answers.

4.3 Results: Question Generation

We evaluate four different aspects of our question generation model: (i) successful control for difficulty, (ii) novelty, (iii) fluency, and (iv) latency.

Difficulty Control To explore whether our question generation model indeed depends on target difficulty and the individual student, we first measure the model's perplexity on a held-out test set of Duolingo questions, compared to permutation baselines. Table 2 (top) shows that perplexity is lower for true student / target difficulty inputs than when either or both of these are permuted. The target difficulty values in this analysis were defined by the LM-DKT model. We can remove this dependence by using the actual student responses from Duolingo: we set the target difficulty to 1 if the student was correct and 0 otherwise. Table 2 (bottom) shows our model prefers questions paired with these "true correctness" targets than paired with random ones.

To evaluate how well our generation model achieves target difficulties, we take 15 unseen students and generate 30 questions for each of 9 input difficulties (0.1-0.9). We then use LM-KT (a well-calibrated proxy for true difficulty) to measure the difficulty of these generated questions for each student. Figure 3 shows that we are able to achieve fine-grained control over target difficulty for both Spanish and French students, with an average Root-Mean Squared Error (RMSE) of **.052** across all students and target difficulties. Adding a sampling penalty (Keskar et al., 2019) increases the variance in difficulty (RMSE .062) in exchange for more novel and diverse questions, as discussed next.

Novelty and Fluency By leveraging a pre-trained language model's ability to manipulate structure, we can generate novel questions not present in the entire Duolingo question set (See Table 3). Across 4,050 questions generated for Spanish learners, we found that with a repetition penalty (Keskar et al., 2019), around 43% of all questions, and 66% of high difficulty ($d = 0.1$) questions, were novel [2]. For French learners, 48% of all and 55% of high difficulty ($d = 0.1$) questions were novel. However, around 3% of generated sentences were judged to be non-fluent [3], although most were still able to be translated (e.g."if i eat some baguettes it breaks."). Without a sampling penalty, the proportion of novel questions drops to about 11 % of questions for French learners (6 % for Spanish learners), yet with far fewer non-fluent examples. Further details and examples of novel and non-fluent generated questions for both Spanish and French learners, are in the Appendix.

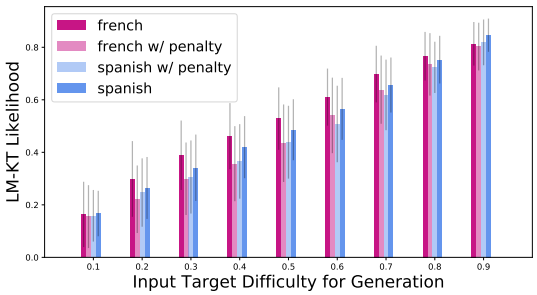


Figure 3: For a random selection of 15 students, our question generator successfully controls for difficulty across a range of 9 target values, evaluated by the LM-KT model. Error bars show standard deviation.

Ablation Type	ppl Spanish	ppl French
LM-DKT Likelihood (0 - 1)		
Ground Truth	**4.33 ±0.20**	**3.86 ±0.09**
Permute Student	6.73 ±0.24	5.11 ±0.41
Permute Difficulty	12.5 ±1.01	7.66 ±0.33
Permute Both	13.1 ±0.43	7.87 ±0.26
Real Student Answers (0 or 1)		
Ground Truth	**17.7 ±1.3**	**9.49 ±.20**
Permute Student	19.75 ±0.43	10.56 ±.60
Permute Difficulty	30.6 ±2.17	13.5 ±0.60
Permute Both	31.3 ±1.49	13.8 ±0.43

Table 2: Perplexity of the question generation model over a held-out evaluation set with ablations.

Latency Positive student experience in online education requires low latency. In about four seconds, our model can generate 30 questions close to a target difficulty. An alternative to question generation is to rank questions from a preexisting pool, according to a target difficulty objective. We compare the quality (RMSE in achieving target difficulty) of the top 30 questions in a pool against the run-time

[2]The CTRL penalty discounts the scores of previously generated tokens, with the HuggingFace Transformers library (Wolf et al., 2020) implementation including tokens provided as part of the prompt. In our setting, this effectively penalizes for generating questions already seen by the student.

[3]We use the language-check Python tool to verify grammar https://pypi.org/project/language-check/.

Difficulty: 0.1 (very hard)	**0.9 (very easy)**
you write letters.	*spoon or tea*
i know about that book.	socks
she reads your letters.	good morning!
those ducks drink water.	a horse
he mixes coffee with water.	*oil against salt*
Difficulty: 0.3 (hard)	**0.7 (easy)**
you drink juice or water	*what dream?*
you drink water	saturday and sunday
accordingly he does it	until tomorrow
can we as a band?	*yes, it is possible!*
during the night	me too

Table 3: Example questions generated by our model for a Spanish learner. *Italic questions* are novel, and do not exist in the Duolingo dataset.

required to rank all questions in the pool, varying its size (Figure 4). On one NVIDIA Titan XP GPU, we find that, averaged across all target difficulties, our question generation model takes half the time to achieve the same quality as pool selection. The gap increases when trying to sample harder questions (d <0.5) – even a pool size of 1000 does not have sufficient difficult questions, likely due to a skew in the Duolingo question set. Additional controls, such as for style or topic, can easily be combined with our generation method, but would make pool selection exponentially more complex.

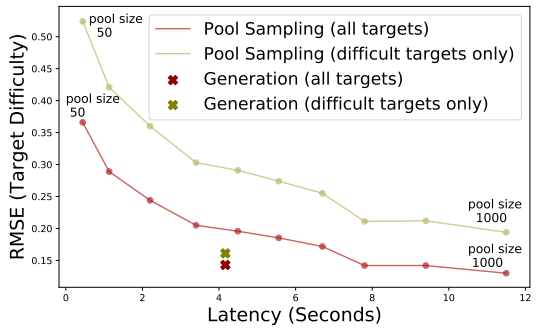


Figure 4: Pool selection (for one student) suffers worse question quality vs. latency trade-off than question generation, especially for sampling difficult questions.

5 Conclusion

Our work is a first step toward showing that sequence-based models combined with domain knowledge, such as pre-trained LMs, can be leveraged for adaptive learning tasks. We show how to use modern LMs to generate novel reverse-translation questions that achieve a target difficulty, allowing adaptive education methods to expand beyond limited question pools.

Limitations of our approach include the compute constraints of large LMs and training data availability. More detailed student data will be crucial to future model development. For instance, while most publicly available education datasets do not include the full student responses (e.g. full translation response in Duolingo), such information could significantly improve the performance of our LM-KT model. Other future directions include exploring non-language domains, such as math or logic exercises, and controlling for auxiliary objectives such as question topic.

Finally, designing appropriate user studies to evaluate our method is a complex yet critical next step to determine its suitability in a real-world education setting. Our techniques allows control for individual student difficulty, but it leaves open the question of optimal curriculum design using difficulty-directed question generation.

6 Broader Impact

Online education platforms can increase the accessibility of high quality educational resources for students around the world. Adaptive techniques that allow for more individualized learning strategies can help such technologies be more inclusive for students who make less-common mistakes or have different prior backgrounds (Lee and Brunskill, 2012). However, our method is subject to biases found in the training data, and careful consideration of using safe and appropriate data is crucial in an education context. Moreover, our specific use of pre-trained LMs relies on the significant progress of NLP tools for English language – further research and development of these tools for other languages can help ensure our method benefits a larger population of students.

7 Acknowledgements

This work was supported in part by the Stanford HAI Hoffman–Yee project "AI Tutors to Help Prepare Students for the 21st Century Workforce". MS was additionally supported by the NSF Graduate Research Fellowship Program under Grant No. DGE 1656518.

References

R. C. Atkinson. 1972. Optimizing the learning of a second-language vocabulary. In *Journal of Experimental Psychology*.

K. C. Bloom and T. J. Shuell. 1981. Effects of massed and distributed practice on the learning and retention of second-language vocabulary. In *The Journal of Educational Research*. Taylor & Francis, Ltd.

A. T. Corbett and J. R. Anderson. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. In *User Modeling and User-Adapted Interaction*. Kluwer Academic Publishers.

Shayan Doroudi and Emma Brunskill. 2019. Fairer but not fair enough on the equitability of knowledge tracing. In *Proceedings of the 9th International Conference on Learning Analytics Knowledge*, LAK19, page 335–339, New York, NY, USA. Association for Computing Machinery.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, Vancouver, Canada. Association for Computational Linguistics.

RK Hambelton and M Jodoin. 2003. Item response theory: models and features.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Anette Hunziker, Yuxin Chen, Oisin Mac Aodha, Manuel Gomez-Rodriguez, Andreas Krause, Pietro Perona, Yisong Yue, and Adish Singla. 2018. Teaching multiple concepts to forgetful learners. *CoRR*, abs/1805.08322.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Jung In Lee and Emma Brunskill. 2012. The impact on individualizing student models on necessary practice opportunities. In *EDM*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Frederic M. Lord. 1980. Applications of item response theory to practical testing problems.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. 2020. Progen: Language modeling for protein generation.

Tong Mu, Shuhan Wang, Erik Andersen, and Emma Brunskill. 2018. Combining adaptivity with progression ordering for intelligent tutoring systems. New York, NY, USA. Association for Computing Machinery.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *NeurIPS*, pages 505–513.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks.

D. Rohrer. 2009. The effects of spacing and mixing practice problems. In *Journal for Research in Mathematics Education*. National Council of Teachers of Mathematics.

B. Settles, C. Brust, E. Gustafson, M. Hagiwara, and N. Madnani. 2018. Second language acquisition modeling. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. ACL.

Burr Settles, Geoffrey T. LaFlair, and Masato Hagiwara. 2020. Machine learning–driven language assessment. *Transactions of the Association for Computational Linguistics*, 8:247–263.

Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. 2020. Saint+: Integrating temporal features for ednet correctness prediction.

David Thissen and Robert J Mislevy. 2000. Testing algorithms.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. 2017. Graph grammar-based controllable generation of puzzles for a learning game about parallel programming. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, New York, NY, USA. Association for Computing Machinery.

David J. Weiss and G. Gage Kingsbury. 1984. Application of computerized adaptive testing to educational problems. In *Journal of Educational Measurement*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A APPENDIX

A.1 Dataset Details

The 2018 Duolingo Shared Task on Second Language Acquisition Modeling (Settles et al., 2018) dataset contains questions and responses for Duolingo users over the first 30 days of learning a second language. The dataset contains three different question types: reverse translate (free response translation of a given prompt in the language they are learning), reverse tap (a selection-based equivalent of reverse translate), and listen, where students listen to a vocal utterance. We focus on the reverse translate question type for English-speaking students learning French and Spanish. The dataset size for French learners (1.2k users) is roughly half the size of that for Spanish learners (2.6k users).

Because the original dataset was intended for per-token error prediction, each question has per-token information that includes whether the student translated the token correctly, as well as Universal Dependencies tags such as part of speech and morphology labels. We use the full question text, rather than individual tokens, for our task, and combine the labels such that if a Duolingo user incorrectly translated one or more tokens in a question, the entire question is marked incorrect. We do not use any additional features.

We use the publicly provided train/dev/test splits from the Shared Task, which are temporally ordered in sequence. We therefore construct student states by tracking user IDs throughout the datasets and appending each new question and response to the current student state. When evaluating our LM-KT model, we use the true responses of preceding questions in the test set to form the student state for a given question. Overall, we find that the dataset is severely imbalanced (as in the original task) - about 30% of questions are answered incorrectly across students studying both French and Spanish.

Finally, we create a held-out set of Duolingo questions for both French and Spanish learners to create the training data for our question generation model. From a set of random student states, we select questions from this set and use a trained LM-KT model to assign the difficulty score. In practice, this held-out set can come from any source, not just Duolingo data.

A.2 Model Training Details

To train both our LM-KT knowledge tracing model and our question generation model, we use the pre-trained OpenAI GPT-2 model from the HuggingFace Transformers library (Wolf et al., 2020). For question generation, we modify the library to add a linear layer and the modified loss function for question generation from Section 3.

We use 1 NVIDIA TitanXP GPU with 12GB of memory available. Because the maximum input sequence length of the GPT-2 model we use is 1024 tokens, we resize all inputs to the last 1024 tokens before training. We report results for an LM-KT model trained for 13k steps with the default batch size of 2 and learning rate of 5e-5, and a Question Generation model trained for 25k steps with the same batch size and learning rate. The total compute time to train both models was 2.5 hours for each language learning task.

A.3 Question Generation Details

For both French and Spanish question generation models, we select 15 students unseen during training and generate 30 questions across 9 difficulties from 0.1 to 0.9, using nucleus sampling (Holtzman et al., 2020) ($p = 0.99$) with a maximum output length of 20 tokens. We also vary a repetition penalty (Keskar et al., 2019) that penalizes for previous tokens (including those in the student state). Lastly, we resize all prompts (student state and target difficulty) to fit into the GPT-2 Model by taking the most recent 1024 tokens, as in training. This is a limitation of our work, as the full student history is not able to be considered for students who have answered a large set of questions.

A.4 Additional Question Generation Outputs

Our question generation model demonstrates the ability to generate novel questions that do not exist in the entire Duolingo question dataset, especially when a sampling penalty is applied to encourage more diverse outputs. However, this comes at a cost to fluency. Below we include a set of outputs generated by our model for 1 Spanish student and 1 French student from the Duolingo dataset, with a target difficulty of $d = 0.1$, and both with and without a repetition penalty. We observe that while applying a penalty results in a far more novel questions generated, several of these are also non-fluent, using a combination of manual judgement and the Python language-check package (https://pypi.org/project/language-check/).

Table 4: Random selection of generated questions for one Spanish learner with for a a target difficulty of $d = 0.1$. *Italic questions* are novel, **bold questions** are judged to be non-fluent.

Spanish (w/ Penalty)	**Spanish (No Penalty)**
accordingly he does it.	*he mixes coffee with milk.*
clean your room or close!	*the cuts are not big.*
clean your room!	*the gallery is enormous.*
he mixes coffee with water.	*the horses are not natural.*
how many elephants eat cheese or fish?	*the men drink a beer.*
i know about that book.	***they probably do not think me.***
october finds him maximum distance from here today!	*we can desk a book.*
please clean your room!	from september to december
please open your bottle or newspaper?	according to you, it is yellow.
she blames us!	clean the mirror.
she reads us lunchtime newspapers.	i do not know it.
she reads your letters.	i read the newspaper.
those ducks drink water.	i want a sandwich without cheese.
we can abandon him.	june starts tomorrow.
what book have they Chosen me so far?	she reads the calendar.
you can control her water.	the plates are not big.
you can establish two properties.	we are following the clue.
your house is very put- pretty!	we drink quickly.
previously on television	we eat strawberries.
you can create the menu.	you can control the water.
you write letters.	you can create the menu.
your hat is gray	you can establish a restaurant.

Table 5: Random selection of generated questions for one French learner with for a a target difficulty of $d = 0.1$. *Italic questions* are novel, **bold questions** are judged to be non-fluent.

French (w/ Penalty)	**French (No Penalty)**
do these children have beans?	*do you have three daughters?*
do they come here often?	*do you like this?*
do we come here often or frequently?	*do you speak french?*
do we have chocolate or water?	*where do the children read?*
do we have coffee here or elsewhere?	do you come here often?
do we have coffee together or onsocks	do you want to dance with me?
do we like to walk distance from one-to two?	some apples, which ones?
do we like to walk together or apart?	corridor or window?
do we speak soon or after tomorrow?	neither do we!
is he chinese or Russian?	you are important.
is he chinese or french?	you are important.
is he sleeping or going out time?	are we going to your place or mine?
map ofis suggests an area.	corridor or window?
otherwise if i want to eat vegetables or fish they regionally cheese, it's meat.	do you have a boyfriend?
some apples of your apple.	do you like to walk?
where do we live today?	neither do we!
where does he go after that jacket?	otherwise, i want a child!
where does she go?	the men are calm and rich.
which ones do not fall victim to be sold?	the parties are in august.
beans and bread	we are reading your letters.
corridor or window?	where do we live?
neither do we!	you eat pork and bread

A Simple Recipe for Multilingual Grammatical Error Correction

Sascha Rothe
Google
rothe@google.com

Jonathan Mallinson
Google
jonmall@google.com

Eric Malmi
Google
emalmi@google.com

Sebastian Krause
Google
bastik@google.com

Aliaksei Severyn
Google
severyn@google.com

Abstract

This paper presents a simple recipe to train state-of-the-art multilingual Grammatical Error Correction (GEC) models. We achieve this by first proposing a language-agnostic method to generate a large number of synthetic examples. The second ingredient is to use large-scale multilingual language models (up to 11B parameters). Once fine-tuned on language-specific supervised sets we surpass the previous state-of-the-art results on GEC benchmarks in four languages: English, Czech, German and Russian. Having established a new set of baselines for GEC, we make our results easily reproducible and accessible by releasing a CLANG-8 dataset.[1] It is produced by using our best model, which we call gT5, to clean the targets of a widely used yet noisy LANG-8 dataset. CLANG-8 greatly simplifies typical GEC training pipelines composed of multiple fine-tuning stages – we demonstrate that performing a single fine-tuning step on CLANG-8 with the off-the-shelf language models yields further accuracy improvements over an already top-performing gT5 model for English.

1 Introduction

Grammatical Error Correction (GEC) is the task of correcting grammatical and other related errors in text. It has been the subject of several modeling efforts in recent years due to its ability to improve grammaticality and readability of user generated texts. This is of particular importance to non-native speakers, children, and individuals with language impairments, who may be more prone to producing texts with grammatical errors.

Modern approaches often view the GEC task as monolingual text-to-text rewriting (Náplava and Straka, 2019; Katsumata and Komachi, 2020; Grundkiewicz et al., 2019) and employ encoder-decoder neural architectures (Sutskever et al., 2014; Bahdanau et al., 2015). These methods typically require large training sets to work well (Malmi et al., 2019) which are scarce especially for languages other than English. One of the largest and most widely used datasets for GEC is the LANG-8 Learner Corpus, which covers 80 languages and has been created by language learners correcting each other's texts.[2] However, the distribution of languages is very skewed, with Japanese and English being the most prevalent languages with over a million ungrammatical-grammatical sentence pairs each, while only ten languages have more than 10,000 sentence pairs each. Additionally, given the uncontrolled nature of the data collection, many of the examples contain unnecessary paraphrasing and erroneous or incomplete corrections.

Limited amounts of suitable training data has led to multiple approaches that propose to generate synthetic training data for GEC (Madnani et al., 2012; Grundkiewicz and Junczys-Dowmunt, 2014; Grundkiewicz et al., 2019; Lichtarge et al., 2019; Awasthi et al., 2019). Although using synthetic data as the first fine-tuning step has been shown to improve model accuracy, it introduces practical challenges that make the development and fair comparison of GEC models challenging: (i) the synthetic methods often require language-specific tuning (e.g. language-specific hyperparameters and spelling dictionaries (Náplava and Straka, 2019)), and; (ii) due to the inability of synthetic data to capture the complete error distribution of the target eval sets, the final model is obtained by following a multi-stage fine-tuning process (Lichtarge et al., 2019, 2020; Omelianchuk et al., 2020). Because of this, carefully picking the learning rates and number of training steps for each of the fine-tuning

[1] CLANG-8 can be found at https://github.com/google-research-datasets/clang8

[2] Corpus collected from https://lang-8.com/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 702–707
August 1–6, 2021. ©2021 Association for Computational Linguistics

stages is required, making it difficult to replicate and build on top of previous best reported models.

The ideas of leveraging self-supervised pre-training and increasing the model size have yielded significant improvements on numerous seq2seq tasks in recent years (Raffel et al., 2019; Xue et al., 2020; Lewis et al., 2020; Song et al., 2019; Chan et al., 2019; Rothe et al., 2020), but these approaches have been applied to GEC to only a limited extent.

In this paper we adopt the mT5 (Xue et al., 2020) as our base model which has already been pre-trained on a corpus covering 101 languages. To adapt the model to the GEC task, we design a fully unsupervised language-agnostic pre-training objective that mimics corrections typically contained in labeled data. We generate synthetic training data by automatically corrupting grammatical sentences, but in contrast to the previous state-of-the-art by Náplava and Straka (2019) for low-resources languages, we use our synthetic pre-training to train a single model on all 101 languages, employing no language-specific priors to remain fully language-agnostic. After pre-training we further fine-tune our model on supervised GEC data for available languages (with data conditions ranging from millions to tens of thousands). Additionally, we explore the effect of scaling up the model size from 60M to 11B parameters. We surpass the previous state-of-the-art results on four evaluated languages: English, Czech, German and Russian.

Fine-tuning and running inference with our largest and most accurate models require multi-GPU/TPU infrastructure. To make the results of our research widely accessible we release a CLANG-8 dataset obtained by using our largest gT5 model to clean up the targets of the frequently used yet noisy LANG-8 dataset. We show that off-the-shelf variants of T5 (Raffel et al., 2019) when fine-tuned only on CLANG-8, outperform those models trained on the original LANG-8 data with and w/o additional fine-tuning data, thus simplifying the complex multi-stage process of training GEC models. Thus CLANG-8 not only allows others to easily train highly competitive GEC models, but it also greatly simplifies GEC training pipeline, basically reducing a multi-step fine-tuning processs to a single fine-tuning step.

Our contributions in this paper are three-fold: (1) We show that a simple language-agnostic pre-training objective can achieve state-of-the-art GEC results when models are scaled up in size; (2) We show the effect model size has on GEC, and; (3) We release a large multilingual GEC dataset based on Lang-8, which allows for state-of-the-art results without additional fine-tuning steps, thus significantly simplifying the training setup.

2 Model

Our model builds on top of mT5 (Xue et al., 2020) a multilingual version of T5 (Raffel et al., 2019) – a Transformer encoder-decoder model which has been shown to achieve state-of-the-art results on a wide range of NLG tasks. mT5 comes in different sizes, however for this work we use *base* (600M parameters) and *xxl* (13B parameters).

2.1 mT5 Pre-training

mT5 has been pre-trained on mC4 corpus, a subset of Common Crawl, covering 101 languages and composed of about 50 billion documents. For details on mC4, we refer the reader to the original paper (Xue et al., 2020). The pre-training objective is based on a span-prediction task, an adaptation of masked-language objective for autoregressive seq2seq models. An example of span prediction:

Input: `A Simple` [x] `Multilingual Grammatical Error` [y]
Target: [x] `Recipe for` [y] `Correction`

All mT5 models were trained for 1M steps on batches of 1024 input sequences with a maximum sequence length of 1024, corresponding to roughly 1T seen tokens. For all of our experiments we use the publicly available mT5 and T5 checkpoints (Section 4 only).

2.2 GEC Pre-training

The span-prediction objective of mT5 does not enable the model to perform GEC without further fine-tuning, as the span-prediction task uses special tokens to indicate where text should be inserted. Another limiting constraint is that mT5 has been trained on paragraphs, not sentences. We therefore split all paragraphs in mC4 corpus into sentences. We corrupt each sentence using a combination of the following operations: a) drop spans of tokens b) swap tokens c) drop spans of characters d) swap characters e) insert characters[3] f) lower-case a word g) upper-case the first

[3]We insert characters from the same passage, thus avoiding to insert character from a different alphabet.

character of a word. An example pair of an original sentence and its corrupted version looks as follows:

Input: `Simple recipe for Multingual Grammatical Correction Error`
Target: `A Simple Recipe for Multilingual Grammatical Error Correction`

We leave about 2% of examples uncorrupted, so the model learns that inputs can also be grammatical. We refrain from using more sophisticated text corruption methods, as these methods would be hard to apply to all 101 languages. For example, Náplava and Straka (2019) perform word substitutions with the entries from ASpell[4] which in turn makes the generation of synthetic data language-specific. Pre-training with this unsupervised objective is done on all languages in the mC4 corpus and not limited to the languages evaluated in this paper.

3 gT5: Large Multilingual GEC Model

Fine-tuning datasets. For English, we fine-tune our pre-trained models on the FCE (Yannakoudakis et al., 2011) and W&I (Bryant et al., 2019a) corpora. For Czech, German, and Russian, we use the AKCES-GEC (Náplava and Straka, 2019), Falko-MERLIN (Boyd, 2018), and RULEC-GEC (Rozovskaya and Roth, 2019) datasets, respectively. Table 1 reports statistics of datasets available for different languages.

lang	Corpus	Train	Dev	Test
EN	FCE, W&I	59,941		
EN	CoNLL-13/-14		1,379	1,312
EN	BEA			4,477
CS	AKCES-GEC	42,210	2,485	2,676
DE	Falko-MERLIN	19,237	2,503	2,337
RU	RULEC-GEC	4,980	2,500	5,000

Table 1: The size of the datasets used to fine-tune gT5.

Training Regime. We experimented with several training setups. All of them build on the mT5 pre-trained models (Section 2.1). We experimented with a) mixing GEC pre-training data (Section 2.2) with fine-tuning data (Section 3), b) mixing pre-training and finetuning examples but annotating them with different prefixes, and c) first using GEC pre-training until convergence and then fine-tuning. While c) is the most computationally expensive approach, it also gave us the best results. GEC pre-training as well as finetuning uses a constant learning rate of 0.001. Pre-training is done until convergence and fine-tuning until exact match accuracy on the development set degrades, which happens after 200 steps or 800k seen examples or 7 epochs.

[4] http://aspell.net

Models	CoNLL-14	BEA test	Czech	German	Russian
Omelianchuk et al.*	66.5	**73.6**	-	-	-
Lichtarge et al.*	**66.8**	73.0	-	-	-
Náplava and Straka	63.40	69.00	80.17	73.71	50.20
Katsumata and Komachi*	63.00	66.10	73.52	68.86	44.36
gT5 base	54.10	60.2	71.88	69.21	26.24
gT5 xxl	65.65	69.83	**83.15**	**75.96**	**51.62**

Table 2: $F_{0.5}$ Scores. Models denoted with * are ensemble models. We used the M^2 scorer for CoNLL-14, Russian, Czech and German, and the ERRANT scorer (Bryant et al., 2019b) for BEA test.

Results. For English, we evaluate on standard benchmarks from CoNLL-14 and the BEA test (Bryant et al., 2019a), while we use CoNLL-13 as the development set (Table 1). For other languages we use the test and development sets associated with their training data. Table 2 shows the results for all languages. We first see that the base model size is inferior to the current state-of-the-art models. This is expected as the model capacity is not enough to cover all 101 languages. We therefore use a larger xxl (11B) model, which produces new state-of-the-art results on all languages except for English. When looking at the development set performance for English, we observed that it had a high variance and the training was over-fitting very quickly. This suggests that train and dev/test set domains are not well aligned for English. In the following Section 4 we further refine our approach, also achieving state-of-the-art results for English.

4 cLang-8: Cleaned Lang-8 Corpus

To be able to distill the knowledge learned by gT5 xxl into smaller, more practical models, we create and release cLang-8, a cleaned version of the popular Lang-8 corpus. As discussed earlier, Lang-8 is a large corpus of texts written by language learners and user-annotated corrections to these texts. However, corrected texts frequently contain unnecessary paraphrasing and erroneous or incomplete corrections – phenomena that hurt the performance of a GEC model trained on this data. For instance, the following source–target pair

	LR	WER	Sub	Del	Ins
LANG-8	98%	15.46	8.85	2.41	4.19
CLANG-8	98%	10.11	5.85	1.35	2.92
CLANG-8-S	99%	01.22	0.64	0.00	0.58

Table 3: Dataset statistics of English LANG-8 and CLANG-8, including sequence **L**ength **R**atio between the source and the target, **W**ord **E**rror **R**ate, which is comprised of **Sub**stitutions, **Del**etions, and **Ins**ertions.

is taken from LANG-8: *"It is cloudy or rainy recently ."* → *"It is It 's been cloudy or and rainy recently ."*

We experiment with two approaches for cleaning the data. First, to create CLANG-8, we generate new targets for LANG-8, disregarding the original targets. We tried using both the unsupervised model, which was trained using the GEC pretraining objective (Section 2.2) and the supervised model (gT5 xxl) (Section 3), but the former did not yield comparable results, so all reported numbers use the supervised model. Second, to create CLANG-8-S, we used the unsupervised and the supervised models to score the original targets, disregarding the lowest scoring 20%, 50%, 70%, or 90% targets. Disregarding 50% was the best performing setup and there was not a significant difference between the supervised and unsupervised model. We therefore report numbers using the unsupervised model disregarding the worst 50% of the targets. Table 3 shows that CLANG-8 moderately reduces the Word Error Rate (WER) between the source and target, with deletions receiving the largest relative reduction, which may suggest that less information from the source sentence is removed. In contrast CLANG-8-S has a significantly lower WER, indicating that the unsupervised model has only kept corrections which are close to the source sentence.

Experiments. To evaluate the effect cleaning LANG-8 has for English, we train two distinct models on this data: T5 (Raffel et al., 2019), a monolingual sequence-to-sequence model, and FELIX (Mallinson et al., 2020), a non-auto-regressive text-editing model.[5] We also tried fine-tuning these models on BEA (i.e. FCE and W&I) after fine-tuning them on CLANG-8, but this did not further improve the scores but slightly decreased them, e.g. 0.43 absolute decrease for BEA test when using T5 base. This can be explained by the fact that the model used to clean the target texts has already been trained on BEA. This suggests that the typical GEC training pipeline where a model is first fine-tuned on LANG-8 and then on BEA can be both simplified and made more accurate by only fine-tuning on CLANG-8.

Finally, we train mT5 models on the German and Russian portions of the CLANG-8 dataset and evaluate these models on the test sets from Table 1.

[5]The FELIXINSERT variant which we use does not employ re-ordering.

Model	#params	Training Data	CoNLL-14	BEA test
SOTA			66.8	73.6
gT5 xxl			65.65	69.83
FELIX	220M	LANG-8	41.63	30.54
FELIX	220M	LANG-8 + BEA	48.75	48.80
FELIX	220M	CLANG-8	**58.21**	**59.05**
T5 base	220M	LANG-8	52.77	59.14
T5 base	220M	LANG-8 + BEA	60.61	67.12
T5 base	220M	CLANG-8	**65.13**	**69.38**
T5 base	220M	CLANG-8-S	58.70	59.95
T5 small	60M	CLANG-8	60.70	65.01
T5 base	220M	CLANG-8	65.13	69.38
T5 large	770M	CLANG-8	66.10	72.06
T5 xl	3B	CLANG-8	67.75	73.92
T5 xxl	11B	CLANG-8	**68.87**	**75.88**

Table 4: $F_{0.5}$ scores on CoNLL-14 and BEA test. Block two and three compare different training data. The last block compares different model sizes for T5.

	LANG-8		CLANG-8	
	base	xxl	base	xxl
PUNCT	68.27	**78.75**	75.51	76.31
DET	63.84	77.31	79.04	**83.88**
PREP	57.09	72.54	74.67	**79.79**
ORTH	72.77	**76.86**	69.23	71.39
SPELL	74.38	84.64	85.83	**88.29**

Table 5: BEA test scores for the top five error types. Bold scores represent the best score for each error type.

Results & Analysis. The results for CoNLL-14 and BEA test benchmarks can be seen in Table 4. For both models and both test datasets, CLANG-8 improves the $F_{0.5}$ score compared to using the original LANG-8 corpus. While CLANG-8-S performs significantly worse than CLANG-8, it still improves over LANG-8. In terms of model size, larger models are consistently better then their smaller siblings. This is even true when comparing xl and xxl, suggesting that there might still be headroom by using models larger than xxl.

In Table 5 we compare error types made on BEA

Model	#params	Training Data	German	Russian
SOTA			73.71	50.20
gT5 xxl			**75.96**	**51.62**
mT5 small	300M	CLANG-8	61.78	17.80
mT5 base	580M	CLANG-8	67.19	25.20
mT5 large	1.2B	CLANG-8	70.14	27.55
mT5 xl	3.7B	CLANG-8	72.59	39.44
mT5 xxl	13B	CLANG-8	74.83	43.52

Table 6: $F_{0.5}$ scores on German and Russian.

test for T5 base and T5 xxl, trained on either LANG-8 or CLANG-8. We see that for both data conditions increasing the model size leads to an increase in performance. Comparing CLANG-8 and LANG-8, shows that CLANG-8 improves on all error types apart from orthographic (ORTH) and punctuation (PUNCT).

In Table 6, we evaluate mT5 trained on the German and Russian portions of the CLANG-8 dataset, which contain 114K and 45K training examples, respectively. We see that for both languages performance increases with the model size, with no indication of slowing, suggesting further headroom for improvement. For German, the xxl model achieves a better score than the previous state-of-the-art, however, it is worse than gT5 xxl. Whereas for Russian, mT5 trained on CLANG-8 does not match state-of-the-art performance. We believe this is in part due to the small size of CLANG-8 in Russian. Additionally, the training data for Russian and German comes from the same dataset as the test data which is not the case for English, making the training data of significantly greater relevance. For German and Russian GEC tasks, where in-domain training data is unavailable, CLANG-8 could have a greater impact.

We release the re-labeled CLANG-8 dataset, which contains 2.4M training examples for English, 114k examples for German, and 45k examples for Russian. The Czech portion of Lang-8 would have resulted in only 2k examples, and as such is excluded.

5 Conclusion

In this paper we report new state-of-the-art results on GEC benchmarks in four languages we studied. Our simple setup relies on a language-agnostic approach to pretrain large multi-lingual language models. To enable the distillation of our largest model into smaller, more efficient models, we released a cleaned version of the LANG-8 dataset, enabling easier and even more accurate training of GEC models.

Acknowledgements

We would like to thank Costanza Conforti, Shankar Kumar, Felix Stahlberg and Samer Hassan for useful discussions as well as their help with training and evaluating the models.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019a. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019b. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. 2019. Kermit: Generative insertion-based modeling for sequences.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *International Conference*

on Natural Language Processing, pages 478–490. Springer.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using pretrained encoder-decoder model. *arXiv preprint arXiv:2005.11849*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jared Lichtarge, Chris Alberti, and Shankar Kumar. 2020. Data weighted training strategies for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 8:634–646.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Exploring grammatical error correction with not-so-crummy machine translation. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53, Montréal, Canada. Association for Computational Linguistics.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Towards Visual Question Answering on Pathology Images

Xuehai He[*1], **Zhuo Cai**[*2], **Wenlan Wei**[3], **Yichen Zhang**[1], **Luntian Mou**[4], **Eric Xing**[5] **and Pengtao Xie**[1]
[1] UC San Diego, [2] Tsinghua University, [3] Wuhan University,
[4] Beijing University of Technology, [5] MBZUAI and CMU
p1xie@eng.ucsd.edu

Abstract

Pathology imaging is broadly used for identifying the causes and effects of diseases or injuries. Given a pathology image, being able to answer questions about the clinical findings contained in the image is very important for medical decision making. In this paper, we aim to develop a pathological visual question answering framework to analyze pathology images and answer medical questions related to these images. To build such a framework, we create PathVQA, a pathology VQA dataset with 32,795 questions asked from 4,998 pathology images. We also propose a three-level optimization framework which performs self-supervised pretraining and VQA finetuning end-to-end to learn powerful visual and textual representations jointly and automatically identifies and excludes noisy self-supervised examples from pretraining. We perform experiments on our created PathVQA dataset and the results demonstrate the effectiveness of our proposed methods. The datasets and code are available at https://github.com/UCSD-AI4H/PathVQA

1 Introduction

Pathology (Levison et al., 2012) studies the causes and effects of diseases or injuries. It underpins every aspect of patient care, such as diagnostic testing, providing treatment advice, preventing diseases using cutting-edge genetic technologies, to name a few. Given a pathology image, being able to answer questions about the clinical findings contained in the image is very important for medical decision-makings.

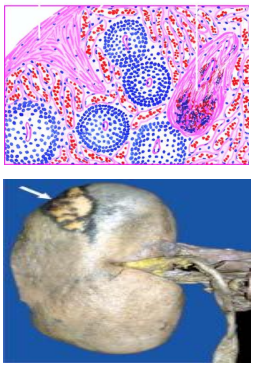
In this paper, we aim to develop a pathological visual question answering framework to analyze pathology images and answer medical questions related to these images. We first need to collect a dataset containing questions about pathology imaging. One possible way to create a pathology VQA dataset is crowdsourcing, which is used successfully for creating general domain VQA datasets (Malinowski and Fritz, 2014; Antol et al., 2015; Ren et al., 2015a; Johnson et al., 2017; Goyal et al., 2017). However, it is much more challenging to build medical VQA datasets than general domain VQA datasets via crowdsourcing. First, medical images such as pathology images are highly domain-specific, which can only be interpreted by well-educated medical professionals. It is rather difficult and expensive to hire medical professionals to help create medical VQA datasets. Second, to create a VQA dataset, one first needs to collect an image dataset. While images in the general domain are pervasive, medical images are very difficult to obtain due to privacy concerns.

To address these challenges, we resort to pathology textbooks, especially those that are freely accessible online, as well as online digital libraries. We extract images and captions from the textbooks and online digital libraries. Given these images, question-answer pairs are created based on image captions. These QA pairs are verified by medical professionals to ensure clinical meaningfulness and correctness. In the end, we created a pathology VQA dataset called PathVQA, which contains 32,795 questions asked from 4,998 pathology images. To our best knowledge, this is the first dataset for pathology VQA.

Given the pathology VQA dataset, the next step is to develop a pathology VQA system, which is also very challenging, due to the following reason. The medical concepts involved in PathVQA are very diverse while the number of question-answer pairs available for training is limited. Learning effective representations of these diverse medical concepts using limited data is technically difficult. Poorly learned representations lead to infe-

[*]Equal Contribution

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 708–718
August 1–6, 2021. ©2021 Association for Computational Linguistics



Q1: What are dilated and congested?
Q2: Are the sinuses dilated and congested?
Q3: Is there increased fibrosis in the red pulp, capsule and the trabeculae?
Q4: Where is increased fibrosis?
Q5: Is gamna-gandy body also seen?

Q1: What is slightly depressed on the surface?
Q2: Where is the wedge-shaped infarct slightly depressed?
Q3: Is the wedge-shaped infarct slightly depressed on the surface?
Q4: What is on the surface?
Q5: What is pale while the margin is haemorrhagic?

Figure 1: Two exemplar images with generated questions. Both images have three types of questions: "what", "where", and "yes/no".

rior VQA performance. To address this challenge, we propose a three-level optimization framework which performs cross-modal self-supervised pretraining (Tan and Bansal, 2019) and VQA finetuning of a pathology image encoder and a question encoder end-to-end to learn powerful visual and textual representations jointly and automatically identifies and excludes noisy self-supervised examples from pretraining. Experiments on our developed PathVQA dataset demonstrates the effectiveness of our proposed methods.

The major contributions of this paper are as follows:

- We create a pathology visual question answering dataset – PathVQA, to foster the research of medical VQA. To our best knowledge, this is the first dataset for pathology VQA.
- We propose a three-level optimization framework which performs cross-modal self-supervised pretraining and VQA finetuning of a pathology image encoder and a question encoder end-to-end to learn powerful visual and textual representations jointly and automatically identifies and excludes noisy self-supervised examples from pretraining.
- On our PathVQA dataset, we demonstrate the effectiveness of our proposed method.

2 Related Work

2.1 Medical VQA Datasets

To our best knowledge, there are two existing datasets for medical visual question answering. The VQA-Med (Abacha et al., 2019) dataset is created on 4,200 radiology images and has 15,292 question-answer pairs. Most of the questions are in multiple-choice (MC) style and can be answered by multi-way classifiers. This makes the difficulty of this dataset significantly lower. VQA-RAD (Lau et al., 2018) is a manually-crafted dataset where questions and answers are given by clinicians on radiology images. It has 3515 questions of 11 types. Our dataset differs from VQA-Med and VQA-RAD in two-fold. First, ours is about pathology while VQA-Med and VQA-RAD (Lau et al., 2018) are both about radiology. Second, our dataset is a truly challenging QA dataset where most of the questions are open-ended while in VQA-Med and VQA-RAD the majority of questions have a fixed number of candidate answers and can be answered by multi-way classification. Besides, the number of questions in our dataset is much larger than that in VQA-Med and VQA-RAD.

2.2 Cross-modal Self-supervised Learning

Cross-modal self-supervised learning learns representations for data with multiple modalities by solving cross-modal auxiliary tasks. VisualBERT (Li et al., 2019) learns representations for images and texts by implicitly aligning elements of a text and regions in an associated image with self-attention. CVLP (Shi et al., 2020) proposes an unbiased contrastive visual-linguistic pretraining approach, which constructs a self-supervised loss based on contrastive learning. ViLBERT (Lu et al., 2019) proposes to pretrain a vision-and-language BERT model through masked multi-modal modeling and alignment tasks, and then transfer the model to visual question answering tasks.

2.3 Data Selection and Data Reweighting

A number of approaches have been proposed for data selection. Ren et al. (2018) proposes a meta learning method to learn the weights of training examples by performing a meta gradient descent step on the weights of the current mini-batch of examples. Shu et al. (2019) propose a method which can adaptively learn an explicit weighting function directly from data.

3 The PathVQA Dataset

The PathVQA dataset consists of 32,795 question-answer pairs generated from 1,670 pathology images collected from two pathology textbooks: "Textbook of Pathology" (Muir et al., 1941) and

Table 1: Frequency of questions in different categories

Question type	Total number and percentage
Yes/No	16,329 (49.8%)
What	13,401 (40.9%)
Where	2,157 (6.6%)
How	595 (1.8%)
How much/many	139 (0.4%)
Why	114 (0.3%)
When	51 (0.2%)
Whose	9 (0.1%)

"Basic Pathology" (Robbins et al., 1981), and 3,328 pathology images collected from the PEIR[1] digital library. The question-answer pairs are generated using a semi-automated pipeline with linguistic rules. Figure 1 shows some examples.

On average, each image has 6.6 questions. The maximum and minimum number of questions for a single image is 14 and 1 respectively. The average number of words per question and per answer is 9.5 and 2.5 respectively. There are eight different categories of questions: what, where, when, whose, how, why, how much/how many, and yes/no. Table 1 shows the number of questions and percentage in each category. The questions in the first 7 categories are open-ended: 16,466 in total and accounting for 50.2% of all questions. The rest are close-ended "yes/no" questions. The questions cover various aspects of visual contents, including color, location, appearance, shape, etc. Such clinical diversity poses great challenges for AI models to solve this pathology VQA problem.

4 Method

We propose a three-level optimization based framework to perform VQA on PathVQA. In our framework, there are three learning stages, which are performed end-to-end jointly. In the first stage, self-supervised learning (He et al., 2019; Tan and Bansal, 2019) is performed to pretrain the image encoder and text encoder. In the second stage, we finetune the image encoder and text encoder on the PathVQA dataset. In the third stage, the trained model is validated on the validation set. In the first stage, we perform cross-modal self-supervised learning (Tan and Bansal, 2019) of an image encoder W and a text encoder T. The image encoder is used to extract visual features of pathology images. The text encoder is used to extract semantic features of questions and answers. Self-supervised learning (He et al., 2019) is an unsupervised representation learning approach where pretext tasks are defined solely based on the input data, and representations are learned by solving these pretext tasks.

There are many ways to construct pretext tasks. In our work, following (Tan and Bansal, 2019), we define a simple yet effective pretext task: in the PathVQA dataset, given a pathology image and a question, judge whether this question is about this image. From the PathVAQ training set D, we create another dataset $D' = \{(x_i, y_i, t_i)\}_{i=1}^{M}$ to perform the SSL task. There are M tuples, each containing a pathology image x from D and a question y from D. t_i is a binary variable where $t_i = 1$ if x and y are from the same training example in D and $t_i = 0$ if otherwise. Given D', we develop a model to map (x_i, y_i) to t_i. In this model, an image encoder is used to encode x_i and a text encoder is used to encode y_i; the concatenation of these two encodings is fed into a linear layer to predict whether the image matches with the question.

In self-supervised learning (He et al., 2019), the labels are typically constructed automatically without human supervision. As a result, they contain a lot of noises. For example, in D', t is determined simply based on whether x and y are from the training example in D. It is totally possible that a question y asked about an image x' is appropriate to be a question for another image x as well if x and x' are pathologically similar. In this case, the correct label t for (x, y) should be 1. However, it is set to 0 in D'. Training the encoders using these noisy and incorrect labels may confuse the encoders and result in poor-quality representations.

To address this problem, we aim to develop a method to automatically identify incorrectly auto-labeled examples in the training data of the SSL task. For each example (x, y, t) in D', we associate a selection variable $a \in [0, 1]$ with it. If a is close to 1, it means this example is correctly labeled; if a is close to 0, it means this example is incorrectly labeled. Let $l(f(x, y; W, T), t)$ denote the SSL loss defined on (x, y, t), where $f(x, y; W, T)$ is the predicted probability that $t = 1$ and $l(\cdot)$ is the cross-entropy loss. We multiply a with $l(f(x, y; W, T), t)$ so that if (x, y, t) is incorrectly

[1] http://peir.path.uab.edu/library/index.php?/category/2

labeled, its loss will be down-weighted to 0 and effectively (x, y, t) is excluded from the SSL pretraining process. In the end, only correctly-labeled examples are used for pretraining the encoders. To this end, in the first stage, we solve the following optimization problem:

$$W^*(A), T^*(A) = \underset{W,T}{\text{argmin}} \sum_{i=1}^{M} a_i l(f(x_i, y_i; W, T), t_i).$$

In this problem, the selection variables $A = \{a_i\}_{i=1}^M$ are fixed (we will discuss how to learn A later on). $\{a_i\}_{i=1}^M$ are used to weigh the losses of individual examples in D. W and T are trained by minimizing the sum of weighted losses. Note that the optimal solutions $W^*(A)$ and $T^*(A)$ are functions of A since $W^*(A)$ and $T^*(A)$ are functions of the loss function, which is a function of A.

In the second stage, we finetune the image encoder and text encoder in the VQA task defined on the PathVQA dataset D. Let V, U, R denote the network weights of the image encoder, text encoder, and QA network respectively. We train V, U, R by minimizing the VQA loss: $\sum_{i=1}^{N^{(\text{tr})}} L(d_i^{(\text{tr})}, V, U, R)$ where $d_i^{(\text{tr})}$ is a training example in D, consisting of an input pathology image, an input question, and an output answer. When training V and U, we encourage them to be close to the optimally trained network weights $W^*(A)$ and $T^*(A)$ of the image and text encoder in the first stage, to transfer the representations learned in the SSL task to the VQA task. The second stage amounts to solving the following optimization problem:

$$\begin{aligned} &V^*(W^*(A)), U^*(T^*(A)), R^* = \\ &\underset{V,U,R}{\text{argmin}} \sum_{i=1}^{N^{(\text{tr})}} L(d_i^{(\text{tr})}, V, U, R) + \\ &\gamma_1 \left\|V - W^*(A)\right\|_2^2 + \gamma_2 \left\|U - T^*(A)\right\|_2^2. \end{aligned} \quad (1)$$

where the L2 losses encourage V and U to be close to $W^*(A)$ and $T^*(A)$. γ_1 and γ_2 are tradeoff parameters. Note that $V^*(W^*(A))$ is a function of $W^*(A)$ since $V^*(W^*(A))$ is a function of $\|V - W^*(A)\|_2^2$ which is a function of $W^*(A)$. Similarly, $U^*(T^*(A))$ is a function of $T^*(A)$.

In the third stage, we apply the optimally trained VQA model including $V^*(W^*(A))$, $U^*(T^*(A))$, and R^* to make predictions on the validation dataset. Then we learn the selection variables A by minimizing the validation loss $\sum_{i=1}^{N^{(\text{val})}} L(d_i^{(\text{val})}, V^*(W^*(A)), U^*(T^*(A)), R^*)$.

Putting all these pieces together, we have the following three-level optimization framework:

$$\begin{aligned} &\min_A \sum_{i=1}^{N^{(\text{val})}} L(d_i^{(\text{val})}, V^*(W^*(A)), U^*(T^*(A)), R^*) \\ &s.t. \quad V^*(W^*(A)), U^*(T^*(A)), R^* = \\ &\quad \text{argmin}_{V,U,R} \sum_{i=1}^{N^{(\text{tr})}} L(d_i^{(\text{tr})}, V, U, R) \\ &\quad\quad +\gamma_1 \left\|V - W^*(A)\right\|_2^2 + \gamma_2 \left\|U - T^*(A)\right\|_2^2 \\ &\quad W^*(A), T^*(A) = \text{argmin}_{W,T} \sum_{i=1}^{M} a_i l(f(x_i, y_i; W, T), t_i) \end{aligned}$$

4.1 VQA Models

Our proposed method can be applied to any VQA method. In this work, we choose two well-established and state-of-the-art VQA methods to perform the study while noting that other VQA methods are applicable as well.

- **Method 1**: In (Tan and Bansal, 2019), a large-scale Transformer (Vaswani et al., 2017) model is built that consists of three encoders: an object relationship encoder, a language encoder, and a cross-modal encoder. The three encoders are built mostly based on two kinds of attention layers — self-attention layers and cross-attention layers. The object relationship encoder and the language encoder are both single-modality encoders. A cross-modal encoder is proposed to learn the connections between vision and language.

- **Method 2**: The method proposed in (Kim et al., 2018) uses a Gated Recurrent Unit (GRU) (Cho et al., 2014) recurrent network and a Faster R-CNN (Ren et al., 2015b) network to embed the question and the image. It extends the idea of co-attention to bilinear attention which considers every pair of multimodal channels.

5 Experiment

5.1 Experimental Settings

Data split We partition the images in the PathVQA dataset along with the associated questions into a training set, validation set, and testing set with a ratio of about 3:1:1. In the PathVQA dataset, the frequencies of question categories are imbalanced. Because of this, during the partition process, we perform sampling to ensure the frequencies of these categories in each set to be consistent. In the end, there are 19,755 question-answer pairs in the training set, 6,279 in the validation set, and 6,761 in the testing set.

Table 2: Accuracy (%), BLEU-n (%), and F1 (%) achieved by different methods. We denote cross-modal SSL on image-question pairs and image-answer pairs as CMSSL-IQ and CMSSL-IA.

Method	Accuracy	BLEU-1	BLEU-2	BLEU-3	F1
Method 1 without image	49.2	50.2	2.8	1.2	9.5
Method 1	57.6	57.4	3.1	1.3	9.9
Method 1 with CMSSL-IQ	58.7	59.0	3.5	2.1	11.0
Method 1 with CMSSL-IQ + three-level optimization framework	**63.4**	**63.7**	**4.1**	**2.5**	**12.2**
Method 1 with CMSSL-IA	58.6	58.9	3.4	2.0	10.3
Method 1 with CMSSL-IA + three-level optimization framework	62.4	62.2	3.6	2.3	12.0
Method 2 without image	46.2	46.5	1.0	0.0	0.8
Method 2	55.1	56.2	3.2	1.2	8.4
Method 2 with CMSSL-IQ	55.9	57.1	3.4	1.4	9.2
Method 2 with CMSSL-IQ + three-level optimization framework	**58.9**	**59.1**	3.8	**1.6**	9.2
Method 2 with CMSSL-IA	55.9	57.1	3.5	1.5	9.2
Method 2 with CMSSL-IA + three-level optimization framework	58.8	**59.1**	**4.0**	**1.6**	**9.4**

Evaluation metrics We perform evaluation using three metrics: 1) accuracy (Malinowski and Fritz, 2014) which measures the percentage of inferred answers that match exactly with the ground-truth using string matching; only exact matches are considered as correct; 2) macro-averaged F1 (Goutte and Gaussier, 2005), which measures the average overlap between the predicted answers and ground-truth, where the answers are treated as bag of tokens; 3) BLEU (Papineni et al., 2002), which measures the similarity of predicted answers and ground-truth by matching n-grams.

5.2 Results

Table 2 shows the VQA performance achieved by different methods. From this table, we make the following observations. **First**, for both Method 1 and Method 2, applying our three-level optimization based framework improves the performance. Our framework learns to identify and remove noisy and erroneous SSL training examples, which can avoid the model to be distorted by such bad-quality examples. **Second**, for both Method 1 and 2, applying cross-modal SSL (CMSSL) methods including CMSSL-IQ and CMSSL-IA improves the performance, which demonstrates the effectiveness of CMSSL. CMSSL uses auxiliary tasks, including judging whether an image matches with a question and judging whether an image matches with an answer, to learn semantic correspondence between image regions and words in questions/answers, which can improve the effectiveness of visual and textual representations for accurate VQA. It also learns image and text encoders by encourages the image and text encoders to solve auxiliary tasks, which reduces the risk of overfitting to the data-deficient VQA task on the small-sized training data.

One may suspect how much information in images is used during the inference of the answers? Could it be possible that the models simply learn the correlations between questions and answers and ignore the images? In light of these concerns, we perform studies where the images are not fed into VQA models and only questions are used as inputs for inferring answers. Table 2 shows the results of not using images ("Method 1/2 without image"). As can be seen, for both Method 1 and 2, ignoring images leads to substantial degradation of performance. This shows that images in our dataset provide valuable information for VQA and PathVQA is a meaningful VQA dataset. The models trained on our datasets are not degenerated to simply capturing the correlation between questions and answers.

6 Conclusion

In this paper, we build a pathology VQA dataset – PathVQA – that contains 32,795 question-answer pairs of 8 categories, generated from 4,998 images. Majority of questions in our dataset are open-ended, posing great challenges for the medical VQA research. Our dataset is publicly available. To address the challenges that the self-supervised training data may contain errors and the effective representations of pathology images and questions are difficult to learn on limited data, we propose a three-level optimization framework to automatically identify and remove problematic SSL training examples and learn sample-efficient visual and textual representations. Experiments on the PathVQA dataset demonstrate the effectiveness of our method.

Acknowledgement

This work is supported by gift funds from Tencent AI Lab and Amazon AWS.

References

Asma Ben Abacha, Sadid A Hasan, Vivek V Datla, Joey Liu, Dina Demner-Fushman, and Henning Müller. 2019. Vqa-med: Overview of the medical visual question answering task at imageclef 2019. In *CLEF2019 Working Notes.*

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. 2014. Zero-shot learning via visual abstraction. In *ECCV*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *HLT-NAACL workshop*.

Zhihao Fan, Zhongyu Wei, Piji Li, Yanyan Lan, and Xuanjing Huang. 2018. A question type driven framework to diversify visual question generation.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NIPS*.

Diederik Kingma and Jimmy Ba. 2014a. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014b. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *ACL*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*.

David Levison, Robin Reid, Alistair D Burt, David J Harrison, and Stewart Fleming. 2012. *Muir's textbook of pathology*. CRC Press.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.

Robert Muir et al. 1941. Text-book of pathology. *Text-Book of Pathology.*, (Fifth Edition).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015a. Image question answering: A visual semantic embedding model and a new dataset. *NIPS*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015b. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Stanley L Robbins, Marcia Angell, and Vinay Kumar. 1981. *Basic pathology*. WB Saunders.

Lei Shi, Kai Shuang, Shijie Geng, Peng Su, Zhengkai Jiang, Peng Gao, Zuohui Fu, Gerard de Melo, and Sen Su. 2020. Contrastive visual-linguistic pretraining. *arXiv preprint arXiv:2007.13135*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1919–1930.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. 2007. The pythy summarization system: Microsoft research at duc 2007. In *Proc. of DUC*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.

C Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *CVPR*.

Table 3: Statistics of the data split.

	Training set	Validation set	Test set
# images	3,021	987	990
# QA pairs	19,755	6,279	6,761

Appendix

A Experimental setup

Table 3 shows dataset split statistics. We implement the methods using PyTorch and perform training on four GTX 1080Ti GPUs.

We basically follow the original model configurations used in (Tan and Bansal, 2019), (Kim et al., 2018), and (Yang et al., 2016). Data augmentation is applied to images, including shifting, scaling, and shearing. From questions and answers in the PathVQA dataset, we create a vocabulary of 4,631 words that have the highest frequencies.

In Method 1, we use the default hyperparameter settings in (Tan and Bansal, 2019). For the text encoder, the hidden size was set to 768. The image features were extracted from the outputs of the Faster-RCNN network, which is pretrained on BCCD[2] – a medical dataset containing blood cells photos, as well as on Visual Genome (Krishna et al., 2017). The initial learning rate was set to 5e-5 with the Adam (Kingma and Ba, 2014a) optimizer used. The batch size was set to 256. The model was trained for 200 epochs. In the SSL pretraining task on Method 1, we train a linear classifier with a dimension of 1,280 to judge whether an image matches with a question. In Method 2, words in questions and answers are represented using GloVe (Pennington et al., 2014) vectors pretrained on general-domain corpora such as Wikipedia, Twitter, etc. The image features are extracted from the outputs of the Faster-RCNN network pretrained on BCCD and Visual Genome. Given an image and a question, the model outputs an answer from a predefined set of answers. The dropout (Srivastava et al., 2014) rate for the linear mapping was set to 0.2 while for the classifier it was set to 0.5. The initial learning rate was set to 0.005 with the Adamax optimizer (Kingma and Ba, 2014b) used. The batch size was set to 512. The model was trained for 200 epochs. In the SSL pretraining task on Method 2, similar to that on Method 1, we train a linear classifier with a dimension of 1,280 to predict whether an image matches with a question. We optimize the selection variables using the Adam optimizer, with an initial learning rate of 0.01. We set γ_1 and γ_2 to 0.3 and 0.7 respectively.

[2]https://public.roboflow.ai/object-detection/bccd

B Dataset Creation

We develop a semi-automated pipeline to generate a pathology VQA dataset from pathology textbooks and online digital libraries. We manually check the automatically-generated question-answer pairs to fix grammar errors. The automated pipeline consists of two steps: (1) extracting pathology images and their captions from electronic pathology textbooks and the Pathology Education Informational Resource (PEIR) Digital Library[3] website; (2) generating questions-answer pairs from captions.

B.1 Extracting Pathology Images and Captions

Given a pathology textbook that is in the PDF format and available online publicly, we use two third-party tools PyPDF2[4] and PDFMiner[5] to extract images and the associated captions therefrom. PyPDF2 provides APIs to access the "Resources" object in each PDF page where the "XObject" gives information about images. PDFMiner allows one to obtain text along with its exact location in a page. To extract image captions from text in each page, we use regular expressions to search for snippets with prefixes of "Fig." or "Figure" followed by figure numbers and caption texts. For a page containing multiple images, we order them based on their locations; the same for the captions. Images and locations are matched based on their order. Given an online pathology digital library such as PEIR, we use two third-party tools Requests[6] and Beautiful Soup[7] to crawl images and the associated captions. Requests is an HTTP library built using Python and provides APIs to send HTTP/1.1 requests. Beautiful Soup generates the 'http.parser' and can access the urls and tags of the images on the website pages. Given a set of urls, we use Requests to read website pages and use Beautiful Soup to find images under the targeted HTML tags including the Content Division element $\langle div \rangle$, the unordered list element $\langle ul \rangle$, and the $\langle li \rangle$ element.

[3]http://peir.path.uab.edu/library/index.php?/category/2
[4]https://github.com/mstamy2/PyPDF2
[5]https://github.com/pdfminer/pdfminer.six
[6]https://requests.readthedocs.io/en/master/
[7]https://www.crummy.com/software/BeautifulSoup/

Table 4: Number of questions in different categories in each set

Dataset	Question types					
	What	Where	How	How much/many	Why	Yes/No
Training set	8083	1316	366	62	71	9804
Validation set	2565	409	108	21	21	3135
Testing set	2753	432	121	18	22	3390

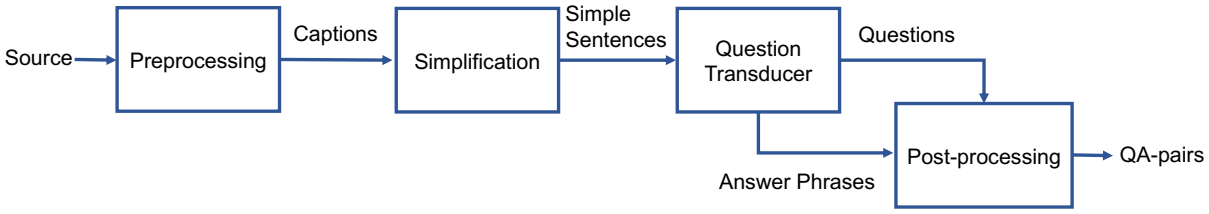


Figure 2: The framework of generating questions from captions

Then we can download images with Requests and write their captions directly to local files. Given the extracted image-caption pairs, we perform post-processing including (1) removing images that are not pathology images, such as flow charts and portraits; (2) correcting erroneous matching between images and captions.

B.2 Question Generation

In this section, we discuss how to semi-automatically generate questions from captions. Figure 2 shows the overall framework. We perform natural language processing of the captions using the Stanford CoreNLP (Klein and Manning, 2003) toolkit, including sentence split, tokenization, part-of-speech (POS) tagging, named entity recognition (NER), constituent parsing, and dependency parsing. Many sentences are long, with complicated syntactic structures. We perform sentence simplification to break a long sentence into several short ones. Given the subjects, verbs, clauses, etc. labeled by POS tagging and syntactic parsing, we rearrange them using the rules proposed in (Toutanova et al., 2007; Dorr et al., 2003) to achieve simplification. Figure 3 shows an example.

Given the POS tags and named entities of the simplified sentences, we generate questions for them: including "when"-type of questions for date and time entities and phrases such as "in/during ... stage/period", "before ...", and "after ..."; "how much/how many"-type of questions for words tagged as numbers; "whose" questions for possessive pronouns (e.g., "its", "their"); "where" questions for location entities and prepositional

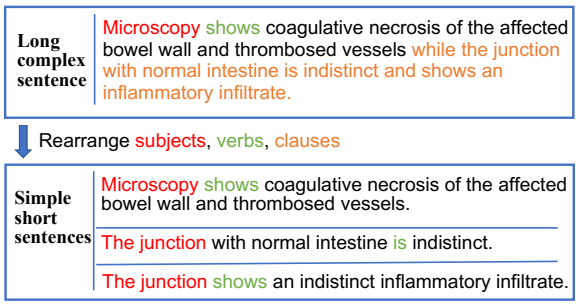


Figure 3: Sentence simplification

phrases starting with "inner", "within", "on the right/left of"; "how" questions for adjective words and phrases starting with "using", "via", "with", and "through", and "what" questions for the remaining noun phrases. Table 5 shows an example for each type of questions.

We use Tregex from Stanford CoreNLP tools (Manning et al., 2014), a tree query language including various relational operators based on the primitive relations of immediate dominance and immediate precedence, to implement the rules (Heilman and Smith, 2009) for transforming declarative sentences (captions) into questions.

To reduce grammatical errors, we avoid generating questions on sentences with adverbial clauses such as "chronic inflammation in the lung, showing all three characteristic histologic features". The question transducer mainly contains three steps. First, we perform the main verb decomposition based on the tense of the verb. For instance, we decompose "shows" to "does show". It is worth noting that for passive sentences with a structure of

Type	Original sentence	Question
What	**The end of the long bone** is expanded in the region of epiphysis.	What is expanded in the region of epiphysis?
Where	The left ventricle is **on the lower right** in this apical four-chamber view of the heart.	Where is the left ventricle in this apical four-chamber view of the heart?
When	**After 1 year of abstinence**, most scars are gone.	When are most scars gone?
How much/How many	**Two** multi-faceted gallstones are present in the lumen.	How many multi-faceted gallstones are present in the lumen?
Whose	The tumor cells and **their** nuclei are fairly uniform, giving a monotonous appearance.	The tumor cells and whose nuclei are fairly uniform, giving a monotonous appearance?
How	The trabecular bone forming the marrow space shows trabeculae **with osteoclastic activity at the margins**.	How does the trabecular bone forming the marrow space show trabeculae?

Table 5: Examples of generated questions for different types

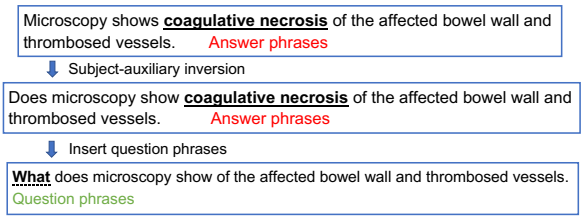


Figure 4: Syntactic transformation

"be+shown/presented/demonstrated", we keep their original forms rather than performing the verb decomposition. Second, we perform subject-auxiliary inversion. We invert the subject and the auxiliary verb in the declarative sentences to form the interrogative sentence. After the inversion, the binary "yes/no" questions are generated. For instance, as shown in Figure 4, the sentence "microscopy shows coagulative necrosis of the affected bowel wall and thrombosed vessels" is inverted to "does microscopy show coagulative necrosis of the affected bowel wall and thrombosed vessels?". To generate questions whose answers are "no", we randomly select a phrase with the same POS tagging from other captions to replace the head words in the original question. For example, we replace "coagulative necrosis" in the sentence "does microscopy show coagulative necrosis of the affected bowel wall and thrombosed vessels" with other noun phrases. Third, we remove the target answer phrases and insert the question phrase obtained previously to generate open-ended questions belonging to types of "what", "where", "when", "whose", "how", and "how much/how many" as shown in Table 5. For instance, we transduce "microscopy shows coagulative necrosis of the affected bowel wall and thrombosed vessels" to "what of the affected bowel wall and thrombosed vessels does microscopy show?" as shown in Figure 4. Given the automatically generated questions which may contain syntactic and semantic errors, we perform post-processing to fix those issues. We manually proofread all questions to correct misspellings, syntactic errors, and semantic inconsistencies. The questions and answers are further cleaned by removing extra spaces and irrelevant symbols. Questions that are too short or vague are removed. Articles appearing at the beginning of answers are stripped.

C Additional Related Works

C.1 VQA datasets

A number of visual question answering datasets have been developed in the general domain. DAQUAR (Malinowski and Fritz, 2014) is built on top of the NYU-Depth V2 dataset (Silberman et al., 2012) which contains RGBD images of indoor scenes. DAQUAR consists of (1) synthetic question-answer pairs that are automatically generated based on textual templates and (2) human-created question-answer pairs produced by five annotators. The VQA dataset (Antol et al., 2015) is developed on real images in MS COCO (Lin et al., 2014) and abstract scene images in (Antol et al., 2014; Zitnick and Parikh, 2013). The question-answer pairs are created by human annotators who are encouraged to ask "interesting" and "diverse" questions. VQA v2 (Goyal et al., 2017) is extended from the VQA (Antol et al., 2015) dataset to achieve more balance between visual and textual information, by collecting complementary images in a way that each question is associated with a pair of similar images with different answers. In the COCO-QA (Ren et al., 2015a) dataset, the question-answer pairs are automatically generated from image captions based on syntactic parsing and linguistic rules. CLEVR (Johnson et al., 2017; Kembhavi et al., 2017) is a dataset developed on rendered images of spatially related objects (including cube, sphere, and cylinder) with different sizes, materials, and colors. The locations and attributes of objects are annotated for each image. The questions are automatically generated from the annotations.

Table 6: Comparison of VQA datasets

	Domain	# images	# QA pairs	Answer type
DAQUAR	General	1,449	12,468	Open
VQA	General	204K	614K	Open/MC
VQA v2	General	204K	1.1M	Open/MC
COCO-QA	General	123K	118K	Open/MC
CLEVR	General	100K	999K	Open
VQA-Med	Medical	4,200	15,292	Open/MC
VQA-RAD	Medical	315	3,515	Open/MC
Ours	Medical	4,998	32,795	Open

The comparison of existing VQA datasets is shown in Table 6. The first five datasets are in the general domain while the last three are in the medical domain. Not surprisingly, the size of general-domain datasets (including the number of images and question-answer pairs) is much larger than that of medical datasets since general-domain images are much more available publicly and there are many qualified human annotators to generate QA pairs on general images. Our dataset is larger than the two medical datasets: VQA-Med and VQA-RAD, and majority of questions in our dataset are open-ended while majority of questions in VQA-Med and VQA-RAD are in multiple-choices style.

C.2 Automatic Construction of Question-Answer Pairs

Existing datasets have used automated methods for constructing question-answer pairs. In DAQUAR, questions are generated with templates, such as "How many {object} are in {image_id}?". These templates are instantiated with ground-truth facts from the database. In COCO-QA, the authors develop a question generation algorithm based on the Stanford syntactic parser (Klein and Manning, 2003), and they form four types of questions—"object", "number", "color", and "location" using hand-crafted rules. In CLEVR, the locations and attributes of objects in each image are fully annotated, based on which the questions are generated by an automated algorithm. Their algorithm cannot be applied to natural images where detailed annotation of objects and scenes are very difficult to obtain. In (Fan et al., 2018), the authors develop a conditional auto-encoder (Kingma and Welling, 2013) model to automatically generate questions from images. To train such a model, image-question pairs are needed, which incurs a chicken-and-egg problem: the goal is to generate questions, but realizing this goal needs generated questions. In VQA-Med, the authors collect medical images along with associated side information (e.g., captions, modalities) from the MedPix[8] database and generate question-answer pairs based on manually-defined patterns in (Lau et al., 2018).

D Number of questions in different categories for training, validation, and test set

For our data split, the number of questions in different categories in each set is shown in Table 4.

[8] https://medpix.nlm.nih.gov

Efficient Text-based Reinforcement Learning by Jointly Leveraging State and Commonsense Graph Representations

Keerthiram Murugesan[I], **Mattia Atzeni**[I,Λ], **Pavan Kapanipathi**[I], **Kartik Talamadupula**[I], **Mrinmaya Sachan**[3], **and Murray Campbell**[I]
[I]IBM Research [Λ]EPFL [3]ETH Zürich
keerthiram.murugesan@ibm.com, atz@zurich.ibm.com,
kapanipa@us.ibm.com, krtalamad@us.ibm.com,
mrinmaya.sachan@inf.ethz.ch, mcam@us.ibm.com

Abstract

Text-based games (TBGs) have emerged as useful benchmarks for evaluating progress at the intersection of grounded language understanding and reinforcement learning (RL). Recent work has proposed the use of external knowledge to improve the efficiency of RL agents for TBGs. In this paper, we posit that to act efficiently in TBGs, an agent must be able to track the state of the game while retrieving and using relevant commonsense knowledge. Thus, we propose an agent for TBGs that induces a graph representation of the game state and jointly grounds it with a graph of commonsense knowledge from ConceptNet. This combination is achieved through *bidirectional knowledge graph attention* between the two symbolic representations. We show that agents that incorporate commonsense into the game state graph outperform baseline agents.

1 Introduction

Text-based games (TBGs) are simulation environments in which an agent interacts with the world purely in the modality of text. TBGs have emerged as key benchmarks for studying how reinforcement learning agents can tackle the challenges of language understanding, partial observability, and action generation in combinatorially large action spaces. One particular text-based gaming environment, TextWorld (Côté et al., 2018), has received significant attention in recent years.

Recent work has shown the need for additional knowledge to tackle the challenges in TBGs. Ammanabrolu and Riedl (2019) proposed handcrafted rules to represent the current state of the game using a state knowledge graph (much like a map of the game). Our own prior work (Murugesan et al., 2021) proposed an extension of TextWorld, called TextWorld Commonsense (TWC), to test agents' ability to use commonsense knowledge while interacting with the world. The hypothesis behind TWC

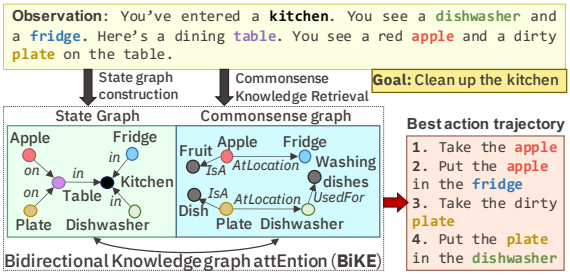


Figure 1: An illustration of a TBG that requires both the state representation of the game as well as the external commonsense knowledge for efficient exploration and learning the best action trajectory. The observation text feeds into the state and commonsense graphs; and the best action trajectory is computed based on information from both graphs.

is that commonsense knowledge allows the agent to understand how current actions might affect future world states; and enable look-ahead planning (Juba, 2016), thus leading to sample-efficient selection of actions at each step and driving the agent closer to optimal performance.

In this paper, we posit that to efficiently act in such text-based gaming environments, an agent must be able to effectively track the state of the game, and use that to jointly retrieve and leverage the relevant commonsense knowledge. For example, commonsense knowledge such as `apple should be placed in the refrigerator` would help the agent to act closer to the optimal behavior; whereas state information like `apple is on the table` would help the agent plan more efficiently. Thus, we propose a technique to: (a) track the state of the game in the form of a symbolic graph that represents the agent's current belief of the state of the world (Ammanabrolu and Hausknecht, 2020a; Adhikari et al., 2020); (b) retrieve the relevant commonsense knowledge from ConceptNet (Speer et al., 2017), and (c) jointly leverage the state graph and the

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 719–725
August 1–6, 2021. ©2021 Association for Computational Linguistics

retrieved commonsense graph. This combined information is then used to select the optimal action. Finally, we demonstrate the performance of our agent against state of the art baseline agents on the TWC Environment.

2 Related Work

Text-based reinforcement learning Text-based games have recently emerged as a promising framework to drive advances in RL research. Prior work has explored text-based RL to learn strategies based on an external text corpus (Branavan et al., 2012) or from textual observations (Narasimhan et al., 2015). In both cases, the text is analyzed and control strategies are learned jointly using feedback from the gaming environment. Zahavy et al. (2018) proposed the Action-Elimination Deep Q-Network (AE-DQN), which learns to classify invalid actions to reduce the action space. The use of the commonsense and state graph in our work has the same goal of down-weighting implausible actions by jointly reasoning over the state of the game and prior knowledge. Recently, Côté et al. (2018) introduced `TextWorld` and Murugesan et al. (2021) proposed *TextWorld Commonsense* (TWC), a text-based gaming environment which requires agents to leverage prior knowledge in order to solve the games. In this work, we build on the agents of Murugesan et al. (2021) and show that prior knowledge and state information are complementary and should be learned jointly.

KG-based state representations A recent line of work in TBGs aims at enhancing generalization performance by using symbolic representations of the agent's belief. Notably, Ammanabrolu and Riedl (2019) proposed *KG-DQN* and Ammanabrolu and Hausknecht (2020b) proposed *KG-A2C*. The idea behind both approaches is to represent the game state as a belief graph. Recently, Adhikari et al. (2020) proposed the graph-aided transformer agent (*GATA*), an approach to construct and update a latent belief graph during planning. Our work integrates these graph-based state representations with a prior commonsense graph that allows the agent to better model the state of the game using prior knowledge.

Sample-efficient reinforcement learning A key challenge for current RL research is low sample efficiency (Kaelbling et al., 1998). To address this problem, there have been few attempts on adding prior or external knowledge to RL approaches. Notably, Murugesan et al. (2020) proposed to use prior knowledge extracted from ConceptNet. Garnelo et al. (2016) proposed *Deep Symbolic RL*, which relies on techniques from symbolic AI as a way to introduce commonsense priors. There has also been work on *policy transfer* (Bianchi et al., 2015) which aims at reusing knowledge gained in different environments. Moreover, *Experience replay* (Wang et al., 2016; Lin, 1992, 1993) provides a framework for how previous experiences can be stored and later reused. In this paper, following Murugesan et al. (2020), we use external KGs as a source of prior knowledge and we combine this knowledge representation with graph-based state modeling in order to allow the agents to act more efficiently.

3 Model & Architecture

TBGs can be framed as partially observable Markov decision processes (POMDPs) (Spaan, 2012) denoted $\langle S, A, O, T, E, r \rangle$, where: S denotes the set of states, A denotes the action space, O denotes the observation space, T denotes the state transition probabilities, E denotes the conditional observation emission probabilities, and $r : S \times A \rightarrow \mathbb{R}$ is the reward function. The observation o_t at time step t depends on the current state. Both observations and actions are rendered in text. The agent receives a reward at every time step t: $r_t = r(o_t, a_t)$, and the agent's goal is to maximize the expected discounted sum of rewards: $\mathbb{E}[\sum_t \gamma^t r_t]$, where $\gamma \in [0, 1]$ is a discount factor.

The high-level architecture of our model contains three major components: (a) the input encoder; (b) a graph-based knowledge extractor; and (c) the action prediction module. The input encoding layers are used to encode the observation o_t at time step t and the list of admissible actions using GRUs (Ammanabrolu and Hausknecht, 2020a). The graph-based knowledge extractor collects relevant knowledge from complementary knowledge sources: the game state, and external commonsense knowledge. We allow information from each knowledge source to guide and direct better representation learning for the other.

Recent efforts have demonstrated the use of primarily two different types of knowledge sources for TextWorld RL Agents. A **State Graph** (SG) captures state information (Ammanabrolu and Riedl, 2019) about the environment represented via a language-based semantic graph. The example in Figure 2 shows that information such as `Apple` →

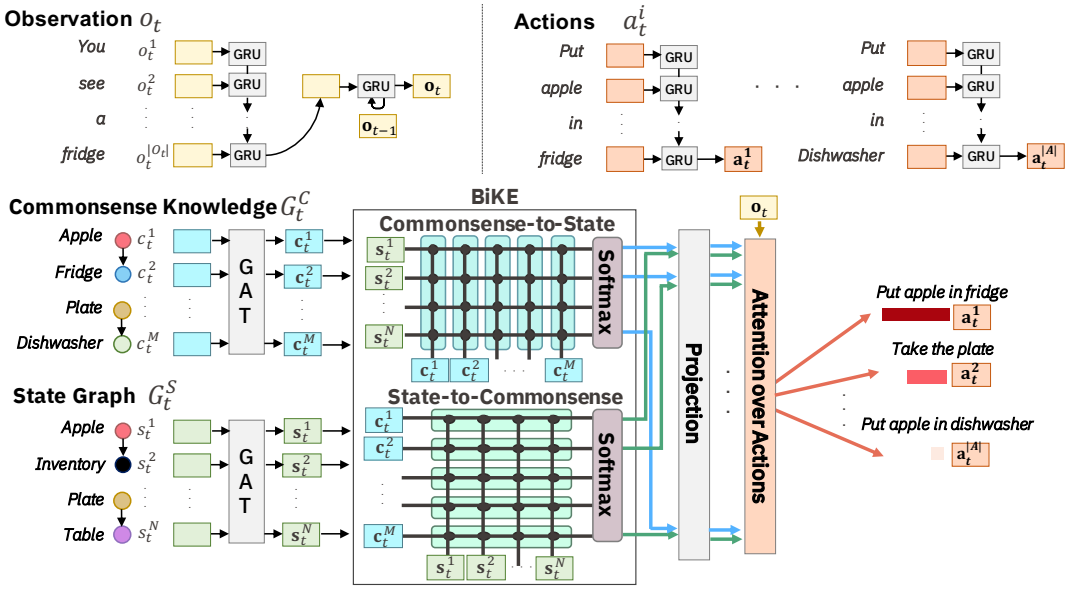


Figure 2: Visualization of our overall approach with BiKE

on → Table is extracted from the textual observations from the environment. Specifically, Ammanabrolu and Riedl (2019) create such knowledge graphs by extracting information using OpenIE (Angeli et al., 2015) and some manual heuristics. A **Commonsense Graph** (CG) captures external commonsense knowledge (Murugesan et al., 2021) between entities (from commonsense knowledge sources such as ConceptNet). We posit that RL agents can make use of information from both these graphs during different sub-tasks, enabling efficient learning. The SG provides the agent with a symbolic way of representing its current perception of the game state, including its understanding of the surroundings. On the other hand, the CG provides the agent with complementary human-like knowledge about what actions make sense in a given state, thus enabling more efficient exploration of the very large natural language based action space.

We combine the state information with commonsense knowledge using a **Bi**directional **K**nowledge-graph attEntion (**BiKE**) mechanism, which re-contextualizes the *state* and *commonsense* graphs based on each other for optimal action trajectories. Figure 2 provides a compact visualization.

4 Knowledge Integration using BiKE

The aforementioned graph-based knowledge extractor produces M entities $(c_t^1, c_t^2, \cdots, c_t^M)$ for the commonsense graph (CG); and N entities $(s_t^1, s_t^2, \cdots, s_t^N)$ for the state graph (SG). Note that the entities extracted for the CG are based on the vocabulary used in ConceptNet, and may not necessarily have the same set of entities as the SG (Figure 1). We embed the extracted entities in both graphs using *Numberbatch* (Liu and Singh, 2004). We then encode these graph representations using a Graph Attention Network (GAT) (Veličković et al., 2018). GAT allows the node entities s_t and c_t within the graphs G_t^S and G_t^C respectively to share information among each other by message passing.

We then integrate sub-graphs extracted from the previous steps to improve the agent's exploration strategy. Inspired from bidirectional attention mechanism in QA (Seo et al., 2016), we use BiKE attention mechanism between G_t^S and G_t^C to fuse the knowledge from these two graphs. The information flow across the graphs allows the model to learn commonsense-aware state graph representations, and state-aware commonsense knowledge graph representations.

To implement this, we compute a graph similarity matrix $\mathtt{S} \in \mathbb{R}^{N \times M}$ across the graph entities to learn a state-to-commonsense graph attention function and a commonsense-to-state graph attention function. $\mathtt{S}_{ij} = f(s_t^i, c_t^j)$ captures how each node s_t^i in the graph G_t^S is linked to a node c_t^j in the other graph G_t^C, and vice versa. Here f is a learnable function that maps s_t^i and c_t^j to a similarity score. This allows us to measure the similarity between (for instance) Apple observed in the state graph and Apple observed in the commonsense graph. We compute the state-to-commonsense graph attention values A by taking a softmax along the rows

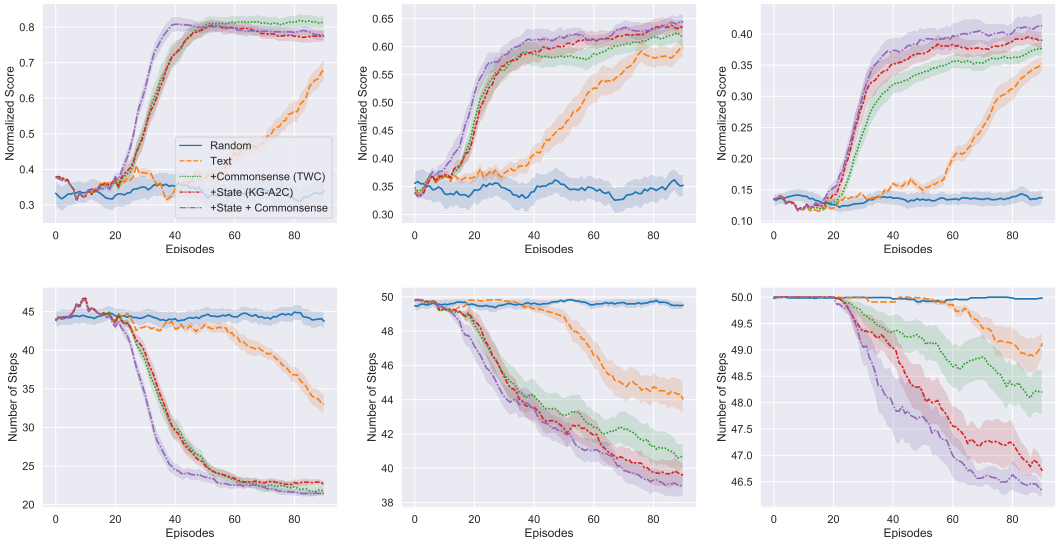


Figure 3: Performance evaluation (showing mean and standard deviation averaged over 3 runs) for the three difficulty levels: Easy (left), Medium (middle), Hard (right) using normalized score and the number of steps taken.

of S: this signifies the attention bestowed by each state graph node on the nodes of the commonsense graph. Similarly, we compute the commonsense-to-state graph attention values $\bar{A}$ by taking a softmax along the columns of S. We capture the relevant knowledge in the commonsense graph G_C^t by updating the state representations $\bar{s}_t^i$. We compute the updated state representation as: $s_{t+1}^i = g(s_t^i, \tilde{s}_t^i, \bar{s}_t^i)$; where $\tilde{s}_t^i = \sum_j A^{ij} c_t^j$, $\bar{s}_t^i = \sum_j A^{ij} \sum_{i'} \bar{A}^{ji'} s_t^{i'}$, and g is a learnable function that maps the concatenated s_t^i, $\tilde{s}_t^i$, and $\bar{s}_t^i$ to an updated state representation. Finally, we use the general attention between the o_t and the state graph entities s_{t+1} to get the state graph representation $\mathbf{g}_{t+1}^S$ (Luong et al., 2015). We perform a similar process for the commonsense-to-state graph attention and obtain the commonsense graph representation: $\mathbf{g}_{t+1}^C$. We select the relevant action by computing an attention over the actions: $h(o_t, a_t^i, \mathbf{g}_{t+1}^S, \mathbf{g}_{t+1}^C)$; where h is a learnable function that projects the concatenation $\langle\, o_t, a_t^i, \mathbf{g}_{t+1}^S, \mathbf{g}_{t+1}^C \,\rangle$ to the attention score for the i^{th} action.

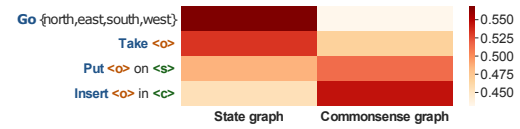
5 Experiments

We generate a set of games with 3 difficulty levels using the TWC (Murugesan et al., 2021) framework: (i) *easy* level, which has 1 room containing 1 to 3 objects; (ii) *medium* level, which has 1 or 2 rooms with 4 or 5 objects; and (iii) *hard* level, a mix of games with a high number of objects (6 or 7 objects in 1 or 2 rooms) or high number of rooms (3 or 4 rooms containing 4 or 5 objects).

We compare 5 text-based RL agents: (a) a text-only agent (**Text**), which selects the best action based only on the encoding of the history of observations; (b) **DRRN** (He et al., 2016; Narasimhan et al., 2015), which relies on the relevance between the observation and action spaces; (c) an agent enhanced with access to an external commonsense knowledge graph (**+Commonsense**) (Murugesan et al., 2021); (d) an agent that, following Ammanabrolu and Hausknecht (2020a), models the state of the world as a symbolic graph (**+State**); and (e) the agent (BiKE) described in Section 3, which relies on both state and commonsense graph representations. The agents are trained over 100 episodes with a 50-step maximum. All policies are learned using Actor-Critic (Mnih et al., 2016).

5.1 Improving Performance with State and Commonsense Knowledge

Figure 3 shows the learning curves for the text-only agent and the agents equipped with state and/or commonsense graph representations at training time. For reference, we also report the performance of an agent that selects a random action at each time step (**Random**). We notice that, overall, agents equipped with either state or commonsense graph representations perform better than their text-only counterparts, both in terms of the number of steps taken and the normalized score. In particular, the BiKE agent outperforms all other agents in all difficulty levels, showing that symbolic state representations and prior commonsense knowledge can be jointly used for better sample efficiency and results. Table 1 shows the performance of the agents on the test set. Following Murugesan et al. (2021), we



(a) Average relevance of the main action templates to the state and commonsense graphs across the *hard* games.

Timestep	t	$\rightarrow t+1$	$\rightarrow t+2$
Room	*Living Room*	→ *Living Room*	→ *Bedroom*
Action Taken	take checkered jumper	→ go west	→ insert checkered jumper into wardrobe
Most relevant graph	*State graph*	*State graph*	*Commonsense graph*
Most relevant nodes	checkered jumper	checkered jumper, exit to west	wardrobe

(b) Example of most relevant graphs and nodes (by action taken) for one example game excerpted from the *hard* difficulty level.

Figure 4: Relevance given to the: (a) state and commonsense graphs; and to (b) their nodes (by action taken).

		Easy		Medium		Hard	
		#Steps	Norm. Score	#Steps	Norm. Score	#Steps	Norm. Score
IN	**Text**	23.83 ± 2.16	0.88 ± 0.04	44.08 ± 0.93	0.60 ± 0.02	49.84 ± 0.38	0.30 ± 0.02
	DRRN	22.08 ± 4.17	0.82 ± 0.06	44.04 ± 1.64	0.59 ± 0.02	49.82 ± 0.61	0.29 ± 0.01
	+Commonsense (TWC)	20.59 ± 5.01	0.89 ± 0.06	42.61 ± 0.65	0.62 ± 0.03	48.45 ± 1.13	0.32 ± 0.04
	+State (KG-A2C)	22.10 ± 2.91	0.86 ± 0.06	41.61 ± 0.37	0.62 ± 0.03	48.00 ± 0.61	0.32 ± 0.00
	+State + Commonsense (BiKE)	18.27 ± 1.13	0.94 ± 0.02	39.34 ± 0.72	0.64 ± 0.02	47.19 ± 0.64	0.34 ± 0.02
OUT	**Text**	29.90 ± 2.92	0.78 ± 0.02	45.90 ± 0.22	0.55 ± 0.01	50.00 ± 0.00	0.20 ± 0.02
	DRRN	29.71 ± 1.81	0.76 ± 0.05	45.18 ± 1.19	0.56 ± 0.02	50.00 ± 0.00	0.21 ± 0.02
	+Commonsense (TWC)	27.74 ± 4.46	0.78 ± 0.07	44.89 ± 1.52	0.58 ± 0.01	50.00 ± 0.00	0.19 ± 0.03
	+State (KG-A2C)	28.34 ± 3.63	0.80 ± 0.07	43.05 ± 2.52	0.59 ± 0.01	50.00 ± 0.00	0.21 ± 0.00
	+State + Commonsense (BiKE)	25.59 ± 1.92	0.83 ± 0.01	41.01 ± 1.61	0.61 ± 0.01	50.00 ± 0.00	0.23 ± 0.02

Table 1: Test-set performance results for within distribution (*IN*) and out-of-distribution (*OUT*) games.

compared our agents on two test sets: (**IN**) uses the same entities as the training set, and (**OUT**) uses entities that were not included in the training set. The experimental results show that the BiKE agent generalizes better than all the baselines across the 3 difficulty levels.

5.2 Qualitative Analysis

From Figure 3 and Table 1, we notice that the **+Commonsense** agent performs better on the *easy* level, whereas the **+State** agent performs better on the *medium* and *hard* levels. This suggests that the state representation can be leveraged to drive exploration and interaction with objects in environments with multiple rooms; whereas prior commonsense knowledge allows the agent to act more efficiently by selecting the appropriate commonsensical locations of different objects. In order to investigate this hypothesis, we computed the average importance given by the agent to the state graph and the commonsense graph when selecting the different action templates shown in Figure 4a. For each action template, the figure shows the normalized attention weight given to the two graphs, averaged across 5 runs of all games in the *hard* difficulty level. Actions requiring information about the goal of the game, like `put` and `insert`, benefit more from attending to the commonsense graph; whereas actions aimed at exploring the environment and collecting objects, like `go` and `take`, benefit more from the state representation.

As further qualitative analysis, we report an example of the most attended nodes and graphs from an excerpt of a game belonging to the *hard* difficulty level in Figure 4b. As noted above, the `take` and `go` actions rely more on the state graph, whereas the `insert` action relies on the commonsense graph. Among the nodes in these graphs, the entities that are finally mentioned in the action receive the highest attention score. This shows how our agent is able to transfer the bidirectional attention over graphs into specific game instances.

6 Conclusion

In this paper, we showed that in order to be sample-efficient in TBGs, agents must be able to jointly track the state of the game and relevant commonsense knowledge. We proposed a technique that models both forms of knowledge as graphs, and combines them using Bidirectional Knowledge-graph attEntion (BiKE). The resulting agent was found to be more sample-efficient than approaches that considered neither or only one of these graphs.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments and suggestions. MS acknowledges support from the Hasler Foundation.

Broader Impact and Discussion of Ethics

While our model is not tuned for any specific real-world application, our method could be used in sensitive contexts such as legal or health-care settings; and it is essential that any work that builds on our approach undertake extensive quality-assurance and robustness testing before using it in their setting. The dataset used in our work does not contain sensitive information to the best of our knowledge.

References

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and William L Hamilton. 2020. Learning dynamic knowledge graphs to generalize on text-based games. *arXiv preprint arXiv:2002.09127*.

Prithviraj Ammanabrolu and Matthew Hausknecht. 2020a. Graph constrained reinforcement learning for natural language action spaces. *arXiv preprint arXiv:2001.08837*.

Prithviraj Ammanabrolu and Matthew J. Hausknecht. 2020b. Graph constrained reinforcement learning for natural language action spaces. In *8th International Conference on Learning Representations, ICLR 2020*.

Prithviraj Ammanabrolu and Mark Riedl. 2019. Playing text-adventure games with graph-based deep reinforcement learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3557–3565.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

Reinaldo AC Bianchi, Luiz A Celiberto Jr, Paulo E Santos, Jackson P Matsuura, and Ramon Lopez de Mantaras. 2015. Transferring knowledge as heuristics in reinforcement learning: A case-based approach. *Artificial Intelligence*, 226:102–121.

SRK Branavan, David Silver, and Regina Barzilay. 2012. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research*, 43:661–704.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2018. Textworld: A learning environment for text-based games. *CoRR*, abs/1806.11532.

Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. 2016. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630.

Brendan Juba. 2016. Integrated common sense learning and planning in pomdps. *The Journal of Machine Learning Research*, 17(1):3276–3312.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134.

Long-Ji Lin. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321.

Long-Ji Lin. 1993. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Pushkar Shukla, Sadhana Kumaravel, Gerald Tesauro, Kartik Talamadupula, Mrinmaya Sachan, and Murray Campbell. 2021. Text-based RL Agents with Commonsense Knowledge: New Challenges, Environments and Baselines. In *The 35th AAAI Conference on Artificial Intelligence*.

Keerthiram Murugesan, Mattia Atzeni, Pushkar Shukla, Mrinmaya Sachan, Pavan Kapanipathi, and Kartik Talamadupula. 2020. Enhancing text-based reinforcement learning agents with commonsense knowledge. *CoRR*, abs/2005.00811.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Matthijs TJ Spaan. 2012. Partially observable markov decision processes. In *Reinforcement Learning*, pages 387–414. Springer.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2016. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*.

Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J Mankowitz, and Shie Mannor. 2018. Learn what not to learn: Action elimination with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3562–3573.

MTVR: Multilingual Moment Retrieval in Videos

Jie Lei **Tamara L. Berg** **Mohit Bansal**
Department of Computer Science
University of North Carolina at Chapel Hill
{jielei, tlberg, mbansal}@cs.unc.edu

Abstract

We introduce MTVR, a large-scale multilingual video moment retrieval dataset, containing 218K English and Chinese queries from 21.8K TV show video clips. The dataset is collected by extending the popular TVR dataset (in English) with paired Chinese queries and subtitles. Compared to existing moment retrieval datasets, MTVR is multilingual, larger, and comes with diverse annotations. We further propose mXML, a multilingual moment retrieval model that learns and operates on data from both languages, via encoder parameter sharing and language neighborhood constraints. We demonstrate the effectiveness of mXML on the newly collected MTVR dataset, where mXML outperforms strong monolingual baselines while using fewer parameters. In addition, we also provide detailed dataset analyses and model ablations. Data and code are publicly available at https://github.com/jayleicn/mTVRetrieval

1 Introduction

The number of videos available online is growing at an unprecedented speed. Recent work (Escorcia et al., 2019; Lei et al., 2020) introduced the Video Corpus Moment Retrieval (VCMR) task: given a natural language query, a system needs to retrieve a short moment from a large video corpus. Figure 1 shows a VCMR example. Compared to the standard text-to-video retrieval task (Xu et al., 2016; Yu et al., 2018), it allows more fine-grained moment-level retrieval, as it requires the system to not only retrieve the most relevant videos, but also localize the most relevant moments inside these videos. Various datasets (Krishna et al., 2017; Hendricks et al., 2017; Gao et al., 2017; Lei et al., 2020) have been proposed or adapted for the task. However, they are all created for a single language (English), though the application could be useful for users speaking other languages as well. Besides, it is also unclear whether the progress and findings in one language generalizes to another language (Bender, 2009). While there are multiple existing multilingual image datasets (Gao et al., 2015; Elliott et al., 2016; Shimizu et al., 2018; Pappas et al., 2016; Lan et al., 2017; Li et al., 2019), the availability of multilingual video datasets (Wang et al., 2019a; Chen and Dolan, 2011) is still limited.

Figure 1: A MTVR example in the Video Corpus Moment Retrieval (VCMR) task. Ground truth moment is shown in *green* box. Colors in the query text indicate whether the words are more related to video (orchid) or subtitle (salmon) or both (orange). The query and the subtitle text are presented in both English and Chinese. The video corpus typically contains thousands of videos, for brevity, we only show 3 videos here.

Therefore, we introduce MTVR, a large-scale, multilingual moment retrieval dataset, with 218K human-annotated natural language queries in two

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 726–734
August 1–6, 2021. ©2021 Association for Computational Linguistics

languages, English and Chinese. MTVR extends the TVR (Lei et al., 2020) dataset by collecting paired Chinese queries and Chinese subtitle text (see Figure 1). We choose TVR over other moment retrieval datasets (Krishna et al., 2017; Hendricks et al., 2017; Gao et al., 2017) because TVR is the largest moment retrieval dataset, and also has the advantage of having dialogues (in the form of subtitle text) as additional context for retrieval, in contrast to pure video context in the other datasets. We further propose mXML, a compact, multilingual model that learns jointly from both English and Chinese data for moment retrieval. Specifically, on top of the state-of-the-art monolingual moment retrieval model XML (Lei et al., 2020), we enforce encoder parameter sharing (Sachan and Neubig, 2018; Dong et al., 2015) where the queries and subtitles from the two languages are encoded using shared encoders. We also incorporate a language neighborhood constraint (Wang et al., 2018; Kim et al., 2020) to the output query and subtitle embeddings. It encourages sentences of the same meaning in different languages to lie close to each other in the embedding space. Compared to separately trained monolingual models, mXML substantially reduces the total model size while improving retrieval performance (over monolingual models) as we show in Section 4. Detailed dataset analyses and model ablations are provided.

2 Dataset

The TVR (Lei et al., 2020) dataset contains 108,965 high-quality English queries from 21,793 videos from 6 long-running TV shows (provided by TVQA (Lei et al., 2018)). The videos are associated with English dialogues in the form of subtitle text. MTVR extends this dataset with translated dialogues and queries in Chinese to support multilingual multimodal research.

2.1 Data Collection

Dialogue Subtitles. We crawl fan translated Chinese subtitles from subtitle sites.[1] All subtitles are manually checked by the authors to ensure they are of good quality and are aligned with the videos. The original English subtitles come with speaker names from transcripts that we map to the Chinese subtitles, to ensure that the Chinese subtitles have the same amount of information as the English version.

[1] https://subhd.tv, http://zimuku.la

QType (%)	Query Examples (in English and Chinese)
video-only (74.2)	Howard places his plate onto the coffee table. 霍华德将盘子放在咖啡桌子上。
sub-only (9.1)	Alexis and Castle talk about the timeline of the murder. 亚历克西斯和卡塞尔谈论谋杀的时间顺序。
video+sub (16.6)	Joey waives his hand when he asks for his food. 乔伊催餐时摆了摆手。

Table 1: MTVR English and Chinese query examples in different query types. The percentage of the queries in each query type is shown in brackets.

Query. To obtain Chinese queries, we hire human translators from Amazon Mechanical Turk (AMT). Each AMT worker is asked to write a Chinese translation of a given English query. Languages are ambiguous, hence we also present the original videos to the workers at the time of translation to help clarify query meaning via spatio-temporal visual grounding. The Chinese translations are required to have the exact same meaning as the original English queries and the translation should be made based on the aligned video content. To facilitate the translation process, we provide machine translated Chinese queries from Google Cloud Translation[2] as references, similar to (Wang et al., 2019b). To find qualified bilingual workers in AMT, we created a qualification test with 5 multiple-choice questions designed to evaluate workers' Chinese language proficiency and their ability to perform our translation task. We only allow workers that correctly answer all 5 questions to participate our annotation task. In total, 99 workers finished the test and 44 passed, earning our qualification. To further ensure data quality, we also manually inspect the submitted results during the annotation process and disqualify workers with poor annotations. We pay workers $0.24 every three sentences, this results in an average hourly pay of $8.70. The whole annotation process took about 3 months and cost approximately $12,000.00.

2.2 Data Analysis

In Table 2, we compare the average sentence lengths and the number of unique words under different part-of-speech (POS) tags, between the two languages, English and Chinese, and between query and subtitle text. For both languages, dialogue subtitles are linguistically more diverse than queries, i.e., they have more unique words in all

[2] https://cloud.google.com/translate

Data	Avg Len	#unique words by POS tags				
		all	verb	noun	adj.	adv.
English						
Q	13.45	15,201	3,015	7,143	2,290	763
Sub	10.78	49,325	6,441	19,223	7,504	1,740
Q+Sub	11.27	52,545	7,151	20,689	8,021	1,976
Chinese						
Q	12.55	34,752	12,773	18,706	1,415	1,669
Sub	9.04	101,018	36,810	53736	4,958	5,568
Q+Sub	9.67	117,448	42,284	62,611	5,505	6,185

Table 2: Comparison of English and Chinese data in MTVR. We show average sentence length, and number of unique tokens by POS tags, for Query (*Q*) and or Subtitle (*Sub*).

categories. This is potentially because the language used in subtitles are unconstrained human dialogues while the queries are collected as declarative sentences referring to specific moments in videos (Lei et al., 2020). Comparing the two languages, the Chinese data is typically more diverse than the English data.[3] In Table 1, we show English and their translated Chinese query examples in Table 1, by query type. In the appendix, we compare MTVR with existing video and language datasets.

3 Method

Our multilingual moment retrieval model mXML is built on top of the Cross-model Moment Localization (XML) (Lei et al., 2020) model, which performs efficient video-level retrieval at its shallow layers and accurate moment-level retrieval at its deep layers. To adapt the monolingual XML model into the multilingual setting in MTVR and improve its efficiency and effectiveness, we apply encoder parameter sharing and neighborhood constraints (Wang et al., 2018; Kim et al., 2020) which encourages the model to better utilize multilingual data to improve monolingual task performance while maintaining smaller model size.

Query and Context Representations. We represent videos using ResNet-152 (He et al., 2016) and I3D (Carreira and Zisserman, 2017) features extracted every 1.5 seconds. We extract language features using pre-trained, then finetuned (on our queries and subtitles) RoBERTa-base (Liu et al., 2019), for English (Liu et al., 2019) and Chinese (Cui et al., 2020), respectively. For queries, we use token-level features. For subtitles, we max-pool the token-level features every 1.5 seconds to align with the video features. We then project the extracted features into a low-dimensional space via a linear layer, and add learned positional encoding (Devlin et al., 2018) after the projection. We denote the resulting video features as $E^v \in \mathbb{R}^{l\times d}$, subtitle features as $E^s_{en} \in \mathbb{R}^{l\times d}$, $E^s_{zh} \in \mathbb{R}^{l\times d}$, and query features as $E^q_{en} \in \mathbb{R}^{l_q\times d}$, $E^q_{zh} \in \mathbb{R}^{l_q\times d}$. l is video length, l_q is query length, and d is hidden size. The subscripts en and zh denote English and Chinese text features, respectively.

[3] The differences might be due to the different morphemes in the languages. E.g., the Chinese word 长发 ('long hair') is labeled as a single noun, but as an adjective ('long') and a noun ('hair') in English (Wang et al., 2019b).

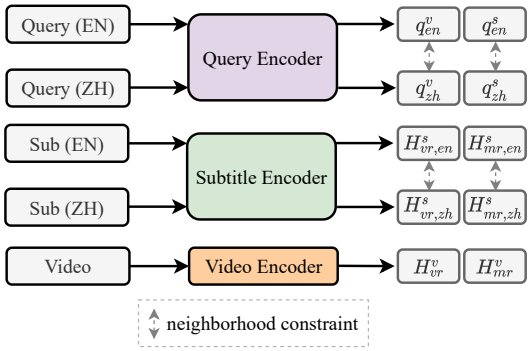


Figure 2: Illustration of mXML's encoding process. Compared to monolingual models, mXML learns from the two languages simultaneously, and allows them to benefit each other via *encoder parameter sharing* and *neighborhood constraints*. We show the detailed encoding process of the model in the appendix (Figure 3).

Encoders and Parameter Sharing. We follow Lei et al. (2020) to use *Self-Encoder* as our main component for query and context encoding. A Self-Encoder consists of a self-attention (Vaswani et al., 2017) layer, a linear layer, and a residual (He et al., 2016) connection followed by layer normalization (Ba et al., 2016). We use a Self-Encoder followed by a modular attention (Lei et al., 2020) to encode each query into two modularized query vectors $\boldsymbol{q}^v_{lang}, \boldsymbol{q}^s_{lang} \in \mathbb{R}^d$ $(lang \in \{en, zh\})$ for video and subtitle retrieval, respectively. For videos, we apply two Self-Encoders instead of a Self-Encoder and a Cross-Encoder as in XML, because we found this modification simplifies the implementation while maintains the performance. We use the outputs from the first and the second Self-Encoder $H^v_{vr,lang}, H^v_{mr,lang} \in \mathbb{R}^{l\times d}$ for video retrieval and moment retrieval. Similarly, we have $H^s_{vr,lang}, H^s_{mr,lang} \in \mathbb{R}^{l\times d}$ for subtitles. All the Self-Encoders are shared across languages, e.g., we use the same Self-Encoder to encode both English and Chinese queries, as illustrated in Figure 2. This parameter sharing strategy greatly reduces the

model size while maintaining or even improving model performance, as we show in Section 4.

Language Neighborhood Constraint. To facilitate stronger multilingual learning, we add neighborhood constraints (Wang et al., 2018; Kim et al., 2020; Burns et al., 2020) to the model. This encourages sentences that express the same or similar meanings to lie close to each other in the embedding space, via a triplet loss. Given paired sentence embeddings $\boldsymbol{e}^i_{en} \in \mathbb{R}^d$ and $\boldsymbol{e}^i_{zh} \in \mathbb{R}^d$, we sample negative sentence embeddings $\boldsymbol{e}^j_{en} \in \mathbb{R}^d$ and $\boldsymbol{e}^k_{zh} \in \mathbb{R}^d$ from the same mini-batch, where $i \neq j, i \neq k$. We use cosine similarity function $\mathcal{S}$ to measure the similarity between embeddings. Our language neighborhood constraint can be formulated as:

$$\mathcal{L}_{nc}=\frac{1}{n}\sum_i[\max(0,\mathcal{S}(\boldsymbol{e}^i_{en},\boldsymbol{e}^k_{zh})-\mathcal{S}(\boldsymbol{e}^i_{en},\boldsymbol{e}^i_{zh})+\Delta) + \max(0,\mathcal{S}(\boldsymbol{e}^j_{en},\boldsymbol{e}^i_{zh})-\mathcal{S}(\boldsymbol{e}^i_{en},\boldsymbol{e}^i_{zh})+\Delta)], \quad (1)$$

where Δ=0.2 is the margin. We apply this constraint on both query and subtitle embeddings, across the two languages, as illustrated in Figure 2. For queries, we directly apply it on the query vectors $\boldsymbol{q}^v_{lang}, \boldsymbol{q}^s_{lang}$. For the subtitle embeddings, we apply it on the embeddings $H^s_{vr,lang}, H^s_{mr,lang}$, after max-pooling them in the temporal dimension.

Training and Inference. During training, we optimize video retrieval scores with a triplet loss, and moment scores with a cross-entropy loss. At inference, these two scores are aggregated together as the final score for video corpus moment retrieval. See appendix for details.

4 Experiments and Results

We evaluate our proposed mXML model on the newly collected MTVR dataset, and compare it with several existing monolingual baselines. We also provide ablation studies evaluating our model design and the importance of each input modality (videos and subtitles).

Data Splits and Evaluation Metrics. We follow TVR (Lei et al., 2020) to split the data into 80% *train*, 10% *val*, 5% *test-public* and 5% *test-private*. We report average recall (R@1) on the Video Corpus Moment Retrieval (VCMR) task. A predicted moment is correct if it has high Intersection-over-Union (IoU) with the ground-truth.

Baseline Comparison. In Table 3, we compare mXML with multiple baseline approaches. Given a natural language query, the goal of video corpus moment retrieval is to retrieve relevant moments from a large video corpus. The methods for this task can be grouped into two categories, (*i*) proposal based approaches (MCN (Hendricks et al., 2017) and CAL (Escorcia et al., 2019)), where they perform video retrieval on the pre-segmented moments from the videos; (*ii*) retrieval+re-ranking methods (MEE (Miech et al., 2018)+MCN, MEE+CAL, MEE+ExCL (Ghosh et al., 2019) and XML (Lei et al., 2020)), where one approach is first used to retrieve a set of videos, then another approach is used to re-rank the moments inside these retrieved videos to get the final moments. Our method mXML also belongs to the retrieval+re-ranking category. Across all metrics and both languages, we notice retrieval+re-ranking approaches achieve better performance than proposal based approaches, indicating that retrieval+re-ranking is potentially better suited for the VCMR task. Meanwhile, our mXML outperforms the strong baseline XML significantly[4] while using few parameters. XML is a monolingual model, where a separate model is trained for each language. In contrast, mXML is multilingual, trained on both languages simultaneously, with parameter sharing and language neighborhood constraints to encourage multilingual learning. mXML prediction examples are provided in the appendix.

Method	#param	English R@1		Chinese R@1	
		IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
Chance	-	0.00	0.00	0.00	0.00
Proposal based					
MCN	6.4M	0.02	0.00	0.13	0.02
CAL	6.4M	0.09	0.04	0.11	0.04
Retrieval + Re-ranking					
MEE+MCN	10.4M	0.92	0.42	1.43	0.64
MEE+CAL	10.4M	0.97	0.39	1.51	0.62
MEE+ExCL	10.0M	0.92	0.33	1.43	0.72
XML	6.4M	7.25	3.25	5.91	2.57
mXML	**4.5M**	**8.30**	**3.82**	**6.76**	**3.20**

Table 3: Baseline comparison on MTVR *test-public* split. mXML achieves better retrieval performance on both languages while using fewer parameters.

Ablations on Model Design. In Table 4, we present our ablation study on mXML. We use '*Baseline*' to denote the mXML model without parameter sharing and neighborhood constraint. Shar-

[4]Statistically significant with p<0.01. We use bootstrap test (Efron and Tibshirani, 1994; Noreen, 1989).

Method	#param	English R@1		Chinese R@1	
		IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
Baseline	6.4M	5.77	2.63	4.7	2.38
+ Share Enc.	**4.5M**	6.09	2.85	4.72	2.25
+ NC (mXML)	**4.5M**	**6.22**	**2.96**	**5.17**	**2.41**

Table 4: mXML ablation study on MTVR *val* split. *Share Enc.* = encoder parameter sharing, *NC* = Neighborhood Constraint. Each row adds an extra component to the row above it.

Model Type	English R@1		Chinese R@1	
	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
Query type: video				
Baseline	5.46	2.53	4.78	2.47
mXML	5.77	2.67	5.14	2.32
Query type: subtitle				
Baseline	4.15	1.97	3.11	1.14
mXML	6.12	3.32	4.05	1.87
Query type: video+subtitle				
baseline	8.02	3.38	5.18	2.62
mXML	8.29	4.09	5.89	3.11

Table 5: Comparison of mXML and the baseline on MTVR *val* set, with breakdown on query types. Both models are trained with video and subtitle as inputs.

ing encoder parameter across languages greatly reduces #parameters while maintaining (Chinese) or even improving (English) model performance. Adding neighborhood constraint does not introduce any new parameters but brings a notable ($p<0.06$) performance gain to both languages. We hypothesize that this is because the learned information in the embeddings of the two languages are complementary (though the sentences in the two languages express the same meaning, their language encoders (Liu et al., 2019; Cui et al., 2020)) are pre-trained differently, which may lead to different meanings at the embedding level. In Table 5, we show a detailed comparison between mXML and its baseline version, by query types. Overall, we notice the mXML perform similarly with *Baseline* in '*video*' queries, but shows a significant performance gain in '*subtitle*' queries, suggesting the parameter sharing and neighborhood constraint are more useful for queries that need more language understanding.

Ablations on Input Modalities. In Table 6, we compare mXML variants with different context inputs, i.e., video or subtitle or both. We report their performance under the three annotated query types, *video*, *sub* and *video+sub*. Overall, the model with both video and subtitle as inputs perform the best. The video model performs much better on the *video* queries than on the *sub* queries, while the subtitle model achieves higher scores on the *sub* queries than the *video* queries.

QType (percentage)	English R@1		Chinese R@1	
	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
Model input: video				
video (74.32%)	4.12	1.89	3.73	1.86
sub (8.85%)	1.97	1.24	1.35	1.04
video+sub (16.83%)	2.67	1.2	2.45	1.15
Model input: subtitle				
video	1.35	0.62	1.11	0.51
sub	6.33	2.9	4.15	1.97
video+sub	6.22	2.62	4.2	2.13
Model input: video+subtitle				
video	5.77	2.67	5.14	2.32
sub	6.12	3.32	4.05	1.87
video+sub	8.29	4.09	5.89	3.11

Table 6: mXML performance breakdown on MTVR *val* set by query types, with different inputs.

In the appendix, we also present results on '*generalization to unseen TV shows*' setup.

5 Conclusion

In this work, we collect MTVR, a new large-scale, multilingual moment retrieval dataset. It contains 218K queries in English and in Chinese from 21.8K video clips from 6 TV shows. We also propose a multilingual moment retrieval model mXML as a strong baseline for the MTVR dataset. We show in experiments that mXML outperforms monolingual models while using fewer parameters.

Acknowledgements

We thank the reviewers for their helpful feedback. This research is supported by NSF Award #1562098, DARPA KAIROS Grant #FA8750-19-2-1004, and ARO-YIP Award #W911NF-18-1-0336. The views contained in this article are those of the authors and not of the funding agency.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Emily M Bender. 2009. Linguistically naïve!= language independent: Why nlp needs linguistic typology. In *Proceedings of the EACL 2009 Workshop*

on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?

Andrea Burns, Donghyun Kim, Derry Wijaya, Kate Saenko, and Bryan A Plummer. 2020. Learning to scale multilingual representations for vision-language tasks. In *ECCV*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. 2019. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*.

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question. In *NIPS*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. 2019. Excl: Extractive clip localization using natural language descriptions. In *NAACL*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan A Plummer. 2020. Mule: Multimodal universal language embedding. In *AAAI*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

Weiyu Lan, Xirong Li, and Jianfeng Dong. 2017. Fluency-guided cross-lingual image captioning. In *ACM MM*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2018. Tvqa: Localized, compositional video question answering. In *EMNLP*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*.

Xirong Li, Chaoxi Xu, Xiaoxu Wang, Weiyu Lan, Zhengxiong Jia, Gang Yang, and Jieping Xu. 2019. Coco-cn for cross-lingual image tagging, captioning, and retrieval. *TMM*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Antoine Miech, Ivan Laptev, and Josef Sivic. 2018. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Nikolaos Pappas, Miriam Redi, Mercan Topkara, Brendan Jou, Hongyi Liu, Tao Chen, and Shih-Fu Chang. 2016. Multilingual visual sentiment concept matching. In *ICMR*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *TACL*.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Nobuyuki Shimizu, Na Rong, and Takashi Miyazaki. 2018. Visual question answering dataset for bilingual image understanding: A study of cross-lingual transfer using attention maps. In *COLING*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. 2018. Learning two-branch neural networks for image-text matching tasks. *TPAMI*.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019a. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019b. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *ECCV*.

A Appendix

Data Analysis. In Table 8 we show a comparison of MTVR with existing moment retrieval datasets and related video and language datasets. Compared to other moment retrieval datasets, MTVR is significantly larger in scale, and comes with query type annotations that allows in-depth analyses for the models trained on it. Besides, it is also the only moment retrieval dataset with multilingual annotations, which is vital in studying the moment retrieval problem under the multilingual context. Compared to the existing multilingual video and language datasets, MTVR is unique as it has a more diverse set of context and annotations, i.e., dialogue, query type, and timestamps.

Training and Inference Details. In Figure 3 we show an overview of the mXML model. We compute video retrieval score as:

$$s^{vr} = \frac{1}{2} \sum_{m \in \{v,s\}} \max\left(\frac{H^m_{vr}}{\|H^m_{vr}\|} \frac{\boldsymbol{q}^m}{\|\boldsymbol{q}^m\|}\right). \quad (2)$$

	English R@1		Chinese R@1	
Setting	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
unseen	1.68	0.79	1	0.54
seen	4.82	2.79	4.18	2.32

Table 7: mXML performance on the MTVR val split *Friends* examples, in both *unseen* and *seen* settings.

The subscript $lang \in \{en, zh\}$ is omitted for simplicity. It is optimized using a triplet loss similar to main text Equation (1). For moment retrieval, we first compute the query-clip similarity scores $S^{q,c} \in \mathbb{R}^l$ as:

$$S^{q,c} = \frac{1}{2}(H^s_{mr}\boldsymbol{q}^s + H^v_{mr}\boldsymbol{q}^v). \quad (3)$$

Next, we apply Convolutional Start-End Detector (ConvSE module) (Lei et al., 2020) to obtain start, end probabilities $P_{st}, P_{ed} \in \mathbb{R}^l$. These scores are optimized using a cross-entropy loss. The single video moment retrieval score for moment $[t_{st}, t_{ed}]$ is computed as:

$$s^{mr}(t_{st}, t_{ed}) = P_{st}(t_{st})P_{ed}(t_{ed}),\ t_{st} \leq t_{ed}. \quad (4)$$

Given a query q_i, the retrieval score for moment [t_{st}:t_{ed}] in video v_j is computed following the aggregation function as in (Lei et al., 2020):

$$s^{vcmr}(v_j, t_{st}, t_{ed}|q_i) = s^{mr}(t_{st}, t_{ed})\exp(\alpha s^{vr}(v_j|q_i)), \quad (5)$$

where α=20 is used to assign higher weight to the video retrieval scores. The overall loss is a simple summation of video and moment retrieval loss across the two languages, and the language neighborhood constraint loss.

Implementation Details. mXML is implemented in PyTorch (Paszke et al., 2017). We use Adam (Kingma and Ba, 2014) with initial learning rate 1e-4, β_1=0.9, β_2=0.999, L2 weight decay 0.01, learning rate warm-up over the first 5 epochs. We train mXML for at most 100 epochs at batch size 128, with early stop based on the sum of R@1 (IoU=0.7) scores for English and Chinese. The experiments are conducted on a NVIDIA RTX 2080Ti GPU. Each run takes around 7 hours.

Generalization to Unseen TV shows. To investigate whether the learned model can be transferred to other TV shows, we conduct an experiment by using the TV show '*Friends*' as an '*unseen*' TV

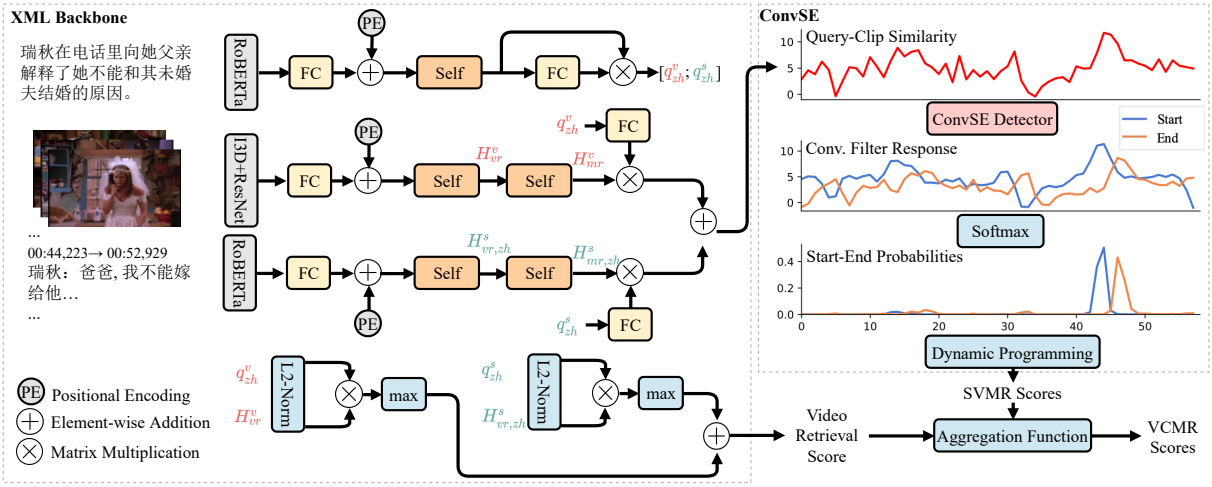


Figure 3: mXML overview. For brevity, we only show the modeling process for a single language (Chinese). The cross-language modifications, i.e., parameter sharing and neighborhood constraint are illustrated in Figure 2. This figure is edited from the Figure 4 in (Lei et al., 2020).

Dataset	Domain	#Q/#videos	Multilingual	Dialogue	QType	Timestamp
QA datasets with temporal annotation						
TVQA (Lei et al., 2018)	TV show	152.5K/21.8K	-	✓	-	✓
How2QA (Li et al., 2020)	Instructional	44K/22K	-	✓	-	✓
Multilingual video description datasets						
MSVD (Chen and Dolan, 2011)	Open	70K/2K	✓	-	-	-
VATEX (Wang et al., 2019b)	Activity	826K/41.3K	✓	-	-	-
Moment retrieval datasets						
TACoS (Regneri et al., 2013)	Cooking	16.2K/0.1K	-	-	-	✓
DiDeMo (Hendricks et al., 2017)	Flickr	41.2K/10.6K	-	-	-	✓
ActivityNet Captions (Krishna et al., 2017)	Activity	72K/15K	-	-	-	✓
CharadesSTA (Gao et al., 2017)	Activity	16.1K/6.7K	-	-	-	✓
How2R (Li et al., 2020)	Instructional	51K/24K	-	✓	-	✓
TVR (Lei et al., 2020)	TV show	109K/21.8K	-	✓	✓	✓
MTVR	TV show	218K/21.8K	✓	✓	✓	✓

Table 8: Comparison of MTVR with related video and language datasets.

show for testing, and train the model on all the other 5 TV shows. For comparison, we also include a model trained on '*seen*' setting, where we use all the 6 TV shows including *Friends* for training. To ensure the models on these two settings are trained on the same number of examples, we downsample the examples in the *seen* setting to match the *unseen* setting. The results are shown in Table 7. We notice our mXML achieves a reasonable performance even though it does see a single example from the TV show *Friends*. Meanwhile, the gap between *unseen* and *seen* settings are still large, we encourage future work to further explore this direction.

Prediction Examples We show mXML prediction examples in Figure 4. We show both Chinese (*top*) and English (*bottom*) prediction examples, and correct (*left*) and incorrect (*right*) examples.

Figure 4: Qualitative examples of mXML. *Top:* examples in Chinese. *Bottom:* examples in English. *Left:* correct predictions. *Right:* incorrect predictions. We show top-3 retrieved moments for each query. salmon bar shows the predictions, green box indicates the ground truth.

Explicitly Capturing Relations between Entity Mentions via Graph Neural Networks for Domain-specific Named Entity Recognition

Pei Chen[1] **Haibo Ding**[2] **Jun Araki**[2] **Ruihong Huang**[1]
[1] Department of Computer Science and Engineering, Texas A&M University
[2] Bosch Research North America
{chenpei, huangrh}@tamu.edu
{Haibo.Ding, Jun.Araki}@us.bosch.com

Abstract

Named entity recognition (NER) is well studied for the general domain, and recent systems have achieved human-level performance for identifying common entity types. However, the NER performance is still moderate for specialized domains that tend to feature complicated contexts and jargonistic entity types. To address these challenges, we propose explicitly connecting entity mentions based on both global coreference relations and local dependency relations for building better entity mention representations. In our experiments, we incorporate entity mention relations by Graph Neural Networks and show that our system noticeably improves the NER performance on two datasets from different domains. We further show that the proposed lightweight system can effectively elevate the NER performance to a higher level even when only a tiny amount of labeled data is available, which is desirable for domain-specific NER.[1]

1 Introduction

Named entity recognition (NER) has been well studied for the general domain, and recent systems have achieved close to human-level performance for identifying a small number of common NER types, such as *Person* and *Organization*, mainly benefiting from the use of Neural Network models (Ma and Hovy, 2016; Yang and Zhang, 2018) and pretrained Language Models (LMs) (Akbik et al., 2018; Devlin et al., 2019). However, the performance is still moderate for specialized domains that tend to feature diverse and complicated contexts as well as a richer set of semantically related entity types (e.g., *Cell*, *Tissue*, *Organ* etc. for the biomedical domain). With these challenges in view, we hypothesize that being aware of the re-occurrences of the same entity as well as semantically related entities will lead to better NER performance for specific domains.

Therefore, we propose to explicitly connect entity mentions in a document that are coreferential or in a tight semantic relation to better learn entity mention representations. Precisely, as shown in Figure 1, we first connect repeated mentions of the same entity even if they are sentences away. For example, the named entity "tumor vasculature" appears both in the *Title* and sentence *S6* but in quite different contexts. Connecting the repeated mentions in a document enables the integration of contextual cues as well as enables consistent predictions of their entity types.

Second, we also connect entity mentions based on sentence-level dependency relations to effectively identify semantically related entities. For example, the two entities in sentence *S3*, "bone marrow" of the type *Multi-tissue Structure* and "endothelial progenitors" of the type *Cell*, are the subject and object of the predicate "contains" respectively in the dependency tree. If the system can reliably predict the type of one entity, we can infer the type of the other entity more easily, knowing that they are closely related on the dependency tree.

We incorporate both relations by using Graph Neural Networks (GNNs), specifically, we use the Graph Attention Networks (GATs) (Velickovic et al., 2018) that have been shown effective for a range of tasks (Sui et al., 2019; Linmei et al., 2019). Empirical results show that our lightweight method can learn better word representations for sequence tagging models and further improve the NER performance over strong LMs-based baselines on two datasets, the AnatEM (Pyysalo and Ananiadou, 2014) dataset from the biomedical domain and the Mars (Wagstaff et al., 2018) dataset from the planetary science domain. In addition, considering the lack of annotations challenge for

[1]The code for the system is available here: https://github.com/brickee/EnRel-G

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 735–742
August 1–6, 2021. ©2021 Association for Computational Linguistics

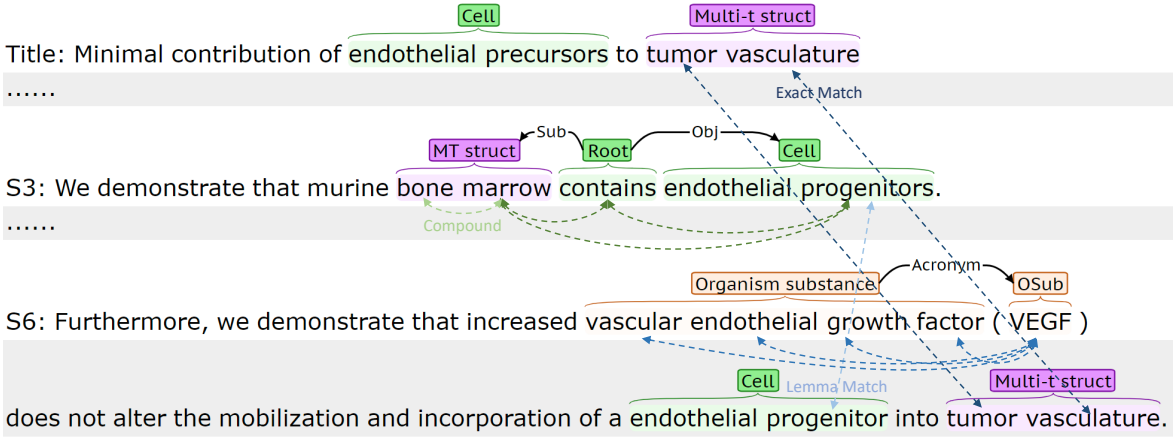


Figure 1: An example of NER with both discourse-level and sentence-level entity relations.

domain-specific NER, we plot learning curves and show that leveraging relations between entity mentions can effectively and consistently improve the NER performance when limited annotations are available.

2 Related Work

NER research has a long history and recent approaches (Yang and Zhang, 2018; Jiang et al., 2019; Jie and Lu, 2019; Li et al., 2020) using Neural Network models like BiLSTM-CNN-CRF (Ma and Hovy, 2016) and contextual embeddings such as BERT (Devlin et al., 2019) and FLAIR (Akbik et al., 2018) have improved the NER performance in the general domain to the human-level. However, the NER performance for specific domains is still moderate due to the challenges of limited annotations and dealing with complicated domain-specific contexts.

We aim to further improve NER performance by considering coreference relations and semantic relations between entity mentions. This is in contrast to the usual way of thinking about NER as an up-stream task conducted before coreference resolution or entity relation extraction. The idea aligns with recent works that conduct joint inferences among multiple information extraction tasks (Miwa and Bansal, 2016; Li et al., 2017; Bekoulis et al., 2018; Luan et al., 2019; Sui et al., 2020; Yuan et al., 2020), including NER, coreference resolution and relation extraction, by mining dependencies among the extractions. However, joint inference approaches require annotations for all the target tasks and aim to improve performance for all the tasks as well, while our lightweight approach aims to improve the performance of the basic NER task requiring no additional annotations (usually unavailable for specific domains).

Our approach is also related to several recent neural approaches for NER that encourage label dependencies among entity mentions. The Pooled FLAIR model (Akbik et al., 2019) proposed a global pooling mechanism to learn word representations. Dai et al. (2019) used a coreference layer with a regularizer to harmonize word representations. Closely related to our work, Qian et al. (2019) used graph neural nets to capture repetitions of the same word as well, but in a denser graph that includes edges between adjacent words and is meant to completely overlay the lower encoding layers. Memory networks (Gui et al., 2020; Luo et al., 2020) were also used to store and refine predictions of a base model by considering repetitions or co-occurrences of words. In addition, dependency relations have been commonly used to connect entities for relation extraction (Zhang et al., 2018; Bunescu and Mooney, 2005), but we aim to better infer the type of an entity by associating it with other closely related entities in a sentence.

3 Model Architecture

Our system with Entity Relation Graphs (EnRelG) mainly contains 5 layers as in Figure 2: an embedding layer, an encoding layer, a GNNs layer, a fusion layer, and a decoding layer.

3.1 Embedding Layer

We choose the BERT-base LM as our embedding layer. For domain-specific datasets, we use BioBERT (Lee et al., 2020) for the biomedical domain and SciBERT (Beltagy et al., 2019) for the planetary science domain. Specifically, for an input

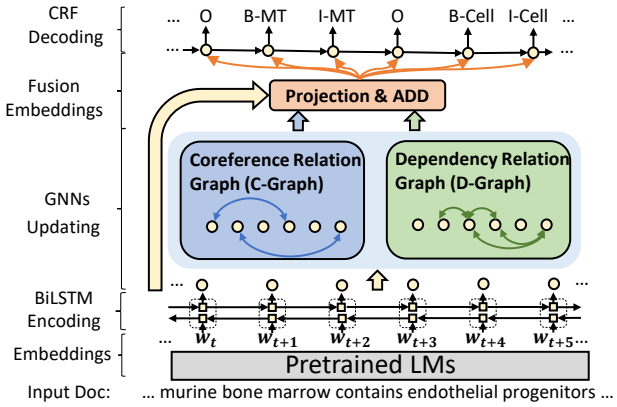


Figure 2: Overall Architecture of the EnRel-G system

document $D = [w_1, w_2, ..., w_n]$ with n words, the BERT model will output a contextual word embeddings matrix $\boldsymbol{E} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_n] \in \mathbb{R}^{n \times d_1}$ with a d_1 dimension vector for each word.

3.2 Encoding Layer

To capture the sequential context information, we use a BiLSTM layer to encode the word embeddings from the BERT model. We concatenate the forward and backward LSTM hidden states as the encoded representations and then obtain embedding matrix $\boldsymbol{E}^{lstm} = BiLSTM(\boldsymbol{E}) \in \mathbb{R}^{n \times d_2}$ with a d_2 dimension vector for each word.

3.3 Graph Neural Networks Layer

For the GNNs layer, we first introduce how to build Entity Relation Graphs using global coreference relations (coreference graph, C-graph) and local dependency relations (dependency graph, D-graph) between entities, and then describe how the GNNs model incorporates them into the word representations.

Coreference Relation Graph For each document, we build a graph $G^C = (\mathcal{V}, \mathcal{A}^C)$ based on coreference relations, in which $\mathcal{V}$ is a set of nodes denoting all the words in a document and $\mathcal{A}^C$ is the adjacency matrix. Specifically, we approximate the entity coreference relations using 3 syntactic coreference clues as in Figure 1: (1) *Exact Match*, two nouns are connected if they are the same, e.g., "tumor vasculature" in both the *Title* and *S6*; (2) *Lemma Match*, two nouns are linked together if they have the same lemma, e.g., "progenitors" and "progenitor" in the *S3* and *S6*; (3) *Acronym Match*, the acronym word is connected to all full expression words, e.g., "VEGF" and "vascular endothelial growth factor" in the *S6*. For each connected node pair (i, j), we set $\mathcal{A}^C_{i,j} = 1$. We also add a self-connection to each node ($\mathcal{A}^C_{i,i} = 1$) to maintain the words' original semantic information.

Dependency Relation Graph We build a Dependency Relation Graphs $G^D = (\mathcal{V}, \mathcal{A}^D)$ for each document based on sentence-level dependency relations. We first parse each sentence using the scispaCy[2] tool and then connect the following word pairs in the dependency tree: (1) *subject head word & object head word & their predicate*, we connect them to enhance the interactions between the entities from the subject and object. e.g., "marrow" and "progenitors" with the predicate "contains" in the *S3*; (2) *compound & head word*, we connect the compounds with their head words because they often both exist in an entity. e.g., the "bone" and "marrow" in the *S3*. Same as before, We set $\mathcal{A}^D_{i,j} = 1$ for each connect pair (i, j), and also add self-connection ($\mathcal{A}^D_{i,i} = 1$) for each node.

Then we update the encoded word embeddings with the entity relations graphs based on GNNs, particularly the GATs. Since nodes represent the words in a document, we initialize the node representations in the graphs from the encoding layer as $\boldsymbol{E}^{lstm} = [\boldsymbol{w}^{lstm}_1, \boldsymbol{w}^{lstm}_2, ..., \boldsymbol{w}^{lstm}_n]$. The graph attention mechanism updates the initial representation of node $\boldsymbol{w}^{lstm}_i$ to $\boldsymbol{w}^{gnn}_i$ by aggregating its neighbors' representations with their corresponding normalized attention scores.

$$\boldsymbol{w}^{gnn}_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha^k_{ij} W^k \boldsymbol{w}^{lstm}_j\right) \tag{1}$$

As in equation (1), and we have K attention heads and concatenate ($\|$) them as the final representation. For head k, we weighted all the adjacent nodes ($\mathcal{N}_i$, obtained from the adjacent matrix $\mathcal{A}$) by W^k and and then aggregate them with the attention score α^k_{ij}. σ is the activation function LeakyReLU. The attention score α^k_{ij} is obtained as followed ($\boldsymbol{a}^T$ is a weight vector):

$$\alpha^k_{ij} = \frac{\exp\left(\sigma\left(\boldsymbol{a}^T\left(W^k \boldsymbol{w}^{lstm}_i \| W^k \boldsymbol{w}^{lstm}_j\right)\right)\right)}{\sum_{z \in \mathcal{N}_i} \exp\left(\sigma\left(\boldsymbol{a}^T\left(W^k \boldsymbol{w}^{lstm}_i \| W^k \boldsymbol{w}^{lstm}_z\right)\right)\right)} \tag{2}$$

For each of the two relation graphs, we use an independent graph attention layer. The output word representations from the two GATs are denoted as: $\boldsymbol{G}^C = [\boldsymbol{w}^{gnn(C)}_1, \boldsymbol{w}^{gnn(C)}_2, ..., \boldsymbol{w}^{gnn(C)}_n] \in \mathbb{R}^{n \times d_3}$ and $\boldsymbol{G}^D = [\boldsymbol{w}^{gnn(D)}_1, \boldsymbol{w}^{gnn(D)}_2, ..., \boldsymbol{w}^{gnn(D)}_n] \in \mathbb{R}^{n \times d_3}$, with d_3 dimension for each word.

[2] https://allenai.github.io/scispacy/

Methods	Datasets	
	AnatEM	Mars
Wagstaff et al. (2018)	–	94.5 / 77.7 / 85.3
NCRF++	83.40±0.34 / 76.96±0.46 / 80.05±0.12	91.28±1.08 / 80.57±0.55 / 85.59±0.23
FLAIR	81.07±0.29 / 75.28±0.57 / 78.06±0.39	90.67±1.02 / 81.45±1.41 / 85.81±0.62
Pooled FLAIR	82.11±0.50 / 77.55±0.40 / 79.76±0.34	87.79±1.31 / 86.57±1.10 / 87.17±0.17
Tuning Bio/SciBERT	83.94±0.40 / 83.12±0.30 / 83.53±0.32	90.93±0.66 / 88.99±1.61 / 89.95±0.64
EnRel-G (C)	84.65±0.67 / 83.69±0.31 / 84.17±0.41	91.21±1.05 / **89.35**±1.76 / 90.27±0.45
EnRel-G (D)	**84.98**±0.83 / 83.50±0.45 / 84.23±0.54	**92.66**±1.16 / 88.03±1.46 / 90.29±0.53
EnRel-G (CD)	84.86±0.50 / **83.96**±0.32 / **84.41**±0.24	92.57±1.00 / 88.65±1.50 / **90.57**±0.47

Table 1: Test results of baselines and our system (Average Precision/Recall/F1 Scores±standard deviation,%)[3]

3.4 Fusion Layer

Similar to Sui et al. (2019), we also use a fusion layer to blend the encoded word embeddings and the GNNs updated word embeddings. We first project these embeddings into the same hidden space using liner transformation and then add them, as in $\boldsymbol{F} = W_N \boldsymbol{E}^{lstm} + W_C \boldsymbol{G}^C + W_D \boldsymbol{G}^D$, where W_N, W_C, W_D are trainable weights. Then we will have a feature matrix $\mathbf{F} \in \mathbb{R}^{n \times d_4}$ for the n words blended with both the sequential context information and global entity relations.

3.5 Decoding Layer

Finally, a Conditional Random Field (CRF) (Lafferty et al., 2001) layer is used to decode the enriched embeddings $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_n]$ into a sequence of labels $y = \{y_1, y_2, ..., y_n\}$. In the training phrase, we optimize the whole model by minimizing the negative log-likelihood loss with respect to gold labels.

4 Experiments[4]

We test our model on two domain-specific datasets: the AnatEM (Pyysalo and Ananiadou, 2014) from the biomedical domain and the Mars (Wagstaff et al., 2018) from the planetary science domain. The AnatEM has annotated 12 types of entities in 1,212 documents with 13,701 entity mentions; the Mars has 117 longer documents with 4,458 entity mentions containing 3 types.

4.1 Baselines

NCRF++ (Yang and Zhang, 2018) is an open-source Neural Sequence Labelling Toolkit. We use the BiLSTM-CNN-CRF structrue as a baseline.

FLAIR (Akbik et al., 2018) is a character-level pretrained LM based on BiLSTM, which has been used in many NER systems (Jiang et al., 2019; Wang et al., 2019). We use the embeddings from it with a BiLSTM-CRF architecture as a baseline.

Pooled FLAIR (Akbik et al., 2019) is an extended version of the FLAIR model with global memory and pooling mechanism for the same word, which helps consistent predictions of coreferential entity mentions. We also use the embeddings from it with a BiLSTM-CRF architecture as a baseline.

Tuning Bio/SciBERT We also use Bio/SciBERT with a BiLSTM-CRF architecture as baselines for the AnatEM/Mars datasets, which do not have the GNNs layer or Fusion layer as compared with our system.

4.2 Results

To alleviate random turbulence, we train all the systems five times using different random seeds and evaluate their average performance on the test sets using the same script[5], as in the Table 1.

We can see that our system with both the global entity coreference and local dependency relations performs the best among all the systems. It improves the average F1 score by 0.88 points (84.41% vs. 83.53%) compared to BioBERT on the AnatEM, and 0.62 points (90.57% vs. 89.95%) compared to SciBERT on the Mars. Further, both the coreference and dependency relations help to improve the NER performance. Specifically, our model with either the coreference or dependency relation graph improves the F1 scores by 0.64 point or 0.7 point on the AnatEM dataset, and by 0.32 point or 0.34 point on the Mars dataset.

[3]Previous systems on the AnatEM dataset either evaluate the NER performance by head match or only evaluate the performance on span identification; therefore, so we do not include their results here.

[4]More details about the datasets, data preprocessing, and model settings can be found in the appendices.

[5]https://github.com/sighsmile/conlleval

Methods	Datasets	
	AnatEM	Mars
Tuning Bio/SciBERT	83.94±0.40 / 83.12±0.30 / 83.53±0.32	90.93±0.66 / 88.99±1.61 / 89.95±0.64
EnRel-G (D) (Key Edges Only)	83.79±0.70 / 83.39±0.39 / 83.59±0.40	91.71±0.63 / 88.30±0.86 / 89.97±0.33
EnRel-G (D) (Compound + Key Edges)	**84.98**±0.83 / 83.50±0.45 / **84.23**±0.54	**92.66**±1.16 / 88.03±1.46 / **90.29**±0.53
EnRel-G (D) (All Modifiers + Key Edges)	84.38±0.72 / **83.83**±0.31 / 84.10±0.40	91.06±1.94 / **89.19**±1.07 / 90.11±0.55
EnRel-G (D) (All Dependency Edges)	84.32±0.36 / 83.52±0.44 / 83.92±0.30	90.71±2.85 / 89.62±1.87 / 90.16±1.23

Table 2: Edge Selection in the Dependency Graph (Average Precision/Recall/F1 Scores±standard deviation,%)

4.3 Learning Curves

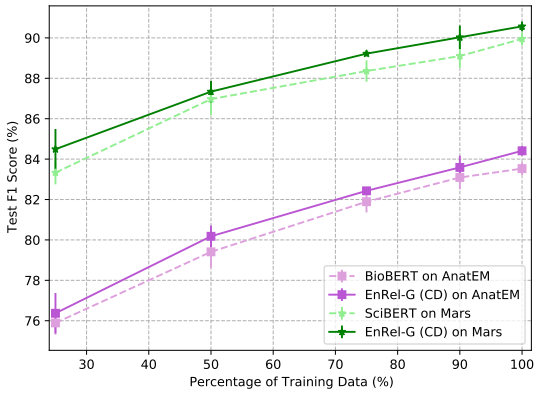


Figure 3: Learning Curves, each point shows the average performance of 5 system runs.

One main limitation of domain-specific NER systems is the lack of annotations, therefore, it is vital to make the best use of labeled data. The learning curves (Figure 3) shows that leveraging the relations between entity mentions can effectively elevate the NER performance to a higher level even when only a tiny amount of labeled data (a quarter of training data) is available, and this is true on both the AnatEM dataset and the Mars dataset.

4.4 Analysis of Computation Cost

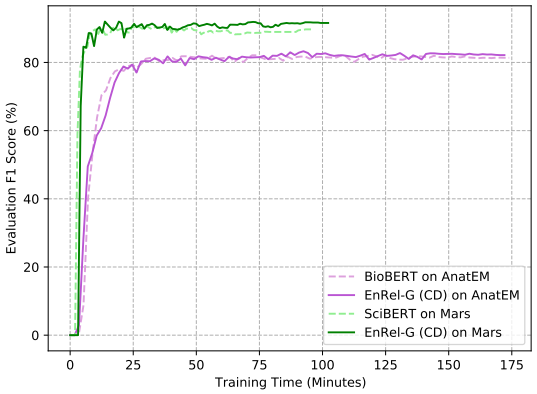


Figure 4: Comparison of Training Time

Although fine-tuning pretrained LMs has improved the performance of many NLP tasks, one limitation is the increase of training time. Therefore, it is important to build computing efficient models based on pretrained LMs. As shown in Figure 4, our model with the GNNs layer does not increase the time cost for fine-tuning the BERT models. The training time of methods with or without the GNNs layer is similar.

4.5 Edge selection in the Dependency Graph

To build the sentence-level dependency graph, we selected only two types of dependency relations: between the subject, object and their predicate (*Key Edges*) and between a compound modifier and its head word. As shown in the Table 2, we also tried to connect all the modifiers with their head word and found that this yields slightly worse performance, and the reason may be that many modifiers other than compounds are not entities themselves. In addition, including all the dependency edges also yields worse performance than using the two selected types of dependency relations, probably for the same reason that many of the nodes in a dependency tree are not parts of entity mentions and many dependency relations do not directly contribute to capturing relations between entities.

5 Conclusion

In this work, we explicitly capture the global coreference and local dependency relations between entity mentions, and use graph neural nets to incorporate the relations to improve domain-specific NER tasks. Experimental results on two datasets show the effectiveness of this lightweight approach. We also find that the selection of entity relations is important to the system performance. Future work may consider about using GNNs to incorporate external knowledge for performance improvement.

Acknowledgments

This work was supported by a gift from Bosch Research and NSF Award IIS-1909255.

References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731.

Zeyu Dai, Hongliang Fei, and Ping Li. 2019. Coreference aware representation learning for neural named entity recognition. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4946–4953. International Joint Conferences on Artificial Intelligence Organization.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tao Gui, Jiacheng Ye, Qi Zhang, Yaqian Zhou, Yeyun Gong, and Xuanjing Huang. 2020. Leveraging document-level label consistency for named entity recognition. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3976–3982. International Joint Conferences on Artificial Intelligence Organization. Main track.

Yufan Jiang, Chi Hu, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2019. Improved differentiable architecture search for language modeling and named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3585–3590, Hong Kong, China. Association for Computational Linguistics.

Zhanming Jie and Wei Lu. 2019. Dependency-guided LSTM-CRF for named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3862–3872, Hong Kong, China. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):1–11.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4823–4832.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *AAAI*, pages 8441–8448.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proceedings of the workshop on detecting structure in scholarly discourse*, pages 27–36.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. GraphIE: A graph-based framework for information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 751–761, Minneapolis, Minnesota. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3821–3831.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *arXiv preprint arXiv:2011.01675*.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kiri Wagstaff, Raymond Francis, Thamme Gowda, You Lu, Ellen Riloff, Karanjeet Singh, and Nina Lanza. 2018. Mars target encyclopedia: Rock and soil composition extracted from the literature. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *International Joint Conference on Artificial Intelligence 2020*, pages 4054–4060. Association for the Advancement of Artificial Intelligence (AAAI).

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Appendices

Appendix A: Dataset Details

The AnatEM (Pyysalo and Ananiadou, 2014) dataset is an extended Anatomical Entity Mention corpus combining both the Anatomical Entity Mention (AnEM) (Ohta et al., 2012) dataset and Multi-level Event Extraction (MLEE) (Pyysalo et al., 2012) corpus. All the documents are selected from PubMed[6] abstracts or full-text papers. AnatEM is manually annotated by biological experts and it has 12 types of entities annotated, namely *Anatomical System, Cancer, Cell, Cellular Component, Developing Anatomical Structure, Immaterial Anatomical Entity, Multi-tissue Structure, Organ, Organism Subdivision, Organism Substance, Pathological Formation, Tissue*. In total, this dataset consists of 1,212 documents and 13,701 entities annotated.

[6] https://pubmed.ncbi.nlm.nih.gov/

Datasets		#Doc	#Words	#Entities	#Words/Doc
AnatEM	Train	606	153,823	6,946	254
	Dev	202	58,785	2,139	291
	Test	404	99,976	4,616	247
	Total	1,212	312,584	13,701	258
Mars	Train	62	99,952	2,431	1,612
	Dev	20	33,743	906	1,687
	Test	35	58,392	1,121	1,668
	Total	117	192,087	4,458	1,642

Table 3: Statistics of the AnatEM and Mars datasets.[7]

Mars is from the scientific literature domain, and it is about planetary science. All documents come from the Lunar and Planetary Science Conference (LPSC)[8], and the entity mentions are annotated manually. It has 3 types of entities: *Element*, *Mineral*, *Target*. The corpus consists of 117 documents. 62 of them are from LPSC 2015 and they are for training and 55 of them are from LPSC 2016 for evaluation. Same as previous work, we divide the 2016 documents into a validation set with 20 documents and a testing set with 35 documents.

Appendix B: Data Preprocessing

We want our model to take advantage of the document-level information, but some of the documents are extremely too long. Moreover, the BERT model also has a limitation of 512 subtokens for input texts. So we need to split the long documents. Besides, the BERT language model needs a big enough batch size (e.g., 16 or 32) to be well fine-tuned, which is also a burden for the GPU memory consumption. In consideration of these restrictions, we limit the max subtoken count of a split document to 128 in the data preprocessing. Future work with more computing resources may try longer input documents.

Moreover, we also add the POS and Dependency Tree information into the data using scispaCy for constructing the Coreference Graph and the Dependency Graph in our model.

Appendix C: Model Settings

For the **NCRF++** baseline, we use one layer of BiLSTM for word sequence representation with 300-dim Glove (Pennington et al., 2014) embeddings, four layers of CNN for character sequence representation with 50-dim random initialized character embeddings, and a CRF layer for inference.

Methods	Optimizer	Learning Rate	Batch Size
NCRF++	SGD	1e-2	10
(pooled) FLAIR	Adam	2e-3	8
Tuning Bio/SciBERT	Adam	5e-5	32
EnRel-G	Adam	5e-5	32

Table 4: Model Settings

For the **FLAIR** and **Pooled FLAIR** baselines, we use the PubMed version (pretrained on the biomedical corpus) for the AnatEM dataset and the general English version (pretrained on the English news articles) for the Mars dataset. Particularly, for the Pooled FLAIR model, we set the *mean* pooling mechanism to calculate the average of embeddings for multiple occurrences of a word, and then use it as the representation for the word.

For the **Tuning BERT** baselines, we use *BioBERT-Base v1.1* for the AnatEM dataset and *SciBERT-scivocab-uncased* for the Mars dataset.

For our **EnRel-G** system, we keep the embeddings layer the same as the Tuning BERT baselines. As for the GNNs layer, we use one layer of the graph attention mechanism with 4 heads, and each head has a hidden dimension of 128.

For the optimization related parameters, as in the Table 4, we mainly use the recommended settings for the baseline models. For our EnRel-G system, we keep the same parameters as in the Tuning BERT baseline for fair comparison.

We train all the systems on a single Nvidia GEFORCE GTX 2080Ti GPU. We set the maximum epoch as 100 and use the best-performed model on the development set to evaluate the test data.

[7] We remove the redundantly annotated entities in the Mars.

[8] https://www.hou.usra.edu/meetings/

Improving Lexically Constrained Neural Machine Translation with Source-Conditioned Masked Span Prediction

Gyubok Lee* **Seongjun Yang*** **Edward Choi**
Graduate School of AI, KAIST
{gyubok.lee,seongjunyang,edwardchoi}@kaist.ac.kr

Abstract

Accurate terminology translation is crucial for ensuring the practicality and reliability of neural machine translation (NMT) systems. To address this, lexically constrained NMT explores various methods to ensure pre-specified words and phrases appear in the translation output. However, in many cases, those methods are studied on general domain corpora, where the terms are mostly uni- and bi-grams (>98%). In this paper, we instead tackle a more challenging setup consisting of domain-specific corpora with much longer n-gram and highly specialized terms. Inspired by the recent success of masked span prediction models, we propose a simple and effective training strategy that achieves consistent improvements on both terminology and sentence-level translation for three domain-specific corpora in two language pairs.

1 Introduction

Despite its recent success in neural machine translation (NMT) (Wu et al., 2016; Johnson et al., 2017; Barrault et al., 2020), delivering correct terms in the translation output is still a vital component for high-quality translation. This concern becomes more salient in domain-specific scenarios, such as in legal documents, where generating correct and consistent terminology is key to ensuring the practicality and reliability of machine translation (MT) systems (Chu and Wang, 2018; Exel et al., 2020).

To address this, lexically constrained NMT works have proposed various methods to preserve terminology in translations as lexical constraints with or without the help of a term dictionary at test time. In most lexically constrained NMT setups, datasets and terms used for training and evaluating the methods are extracted from WMT news corpora (Dinu et al., 2019; Susanto et al., 2020; Chen et al., 2020). Since the terms, regardless of their source, can only be utilized as long as they exist in the corpus, the term coverage solely depends on the choice of the corpus. By analyzing the previous setups carefully, we discover that the terms found in WMT are mostly uni- or bi-grams (see Figure 1) and highly colloquial (see Table 1 for the top 10 most frequent terms). These leave the question of whether the previous methods are effective in domain-specific scenarios where accurate terminology translation is truly vital.

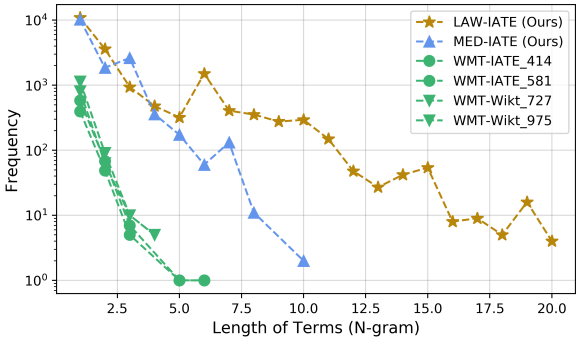


Figure 1: The frequency of terms sorted by n-gram between Dinu et al. (2019)'s and our test splits. While the terms in WMT De-En are mostly uni- or bi-grams, our setup contains heavy-tailed n-gram distributions with more quantity and diversity in terminology.

In this paper, inspired by the recent masked span prediction models, which have demonstrated improved representation learning capability of contiguous words (Song et al., 2019b; Joshi et al., 2019; Lewis et al., 2020; Raffel et al., 2020), we propose a simple yet effective training scheme to improve terminology translation in highly specialized domains. We specifically select two highly specialized domains (*i.e.*, law and medicine) which contain domain-specific terminologies to address more challenging and realistic setups, in addition to applying it to both typologically similar and dissimilar pairs of languages (German-English (De→En) and Korean-English (Ko→En)). Thanks to its sim-

* equal contributions

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 743–753
August 1–6, 2021. ©2021 Association for Computational Linguistics

plicity, the proposed method is compatible with any autoregressive Transformer-based model, including ones capable of utilizing term dictionaries at training or test time. In domain-specific setups where longer n-gram terms are pervasive, our method demonstrates improved performance over the standard maximum likelihood estimation (MLE) approach in terms of terminology and sentence-level translation quality. Our code and datasets are available at https://github.com/wns823/NMT_SSP.

2 Background

Lexically constrained NMT We could group lexically constrained NMT methods into two streams: *hard* and *soft*. The hard approaches aim to force all terminology constraints to appear in the generated output. The methods include replacing constraints (Crego et al., 2016), constrained decoding (Hokamp and Liu, 2017; Chatterjee et al., 2017; Post and Vilar, 2018; Hasler et al., 2018), and additional attention heads for external supervision (Song et al., 2020). Although those approaches are reliable and widely used in practice, they typically require a pre-specified term dictionary and an extra candidate selection module if there are multiple matching candidates for a single term (see caption in Table 2).

Several soft methods address this problem without the help of a term dictionary, one of which is training on both constraint pseudo-labeled (with statistical MT) and unlabeled data (Song et al., 2019a). More recently, Susanto et al. (2020) and Chen et al. (2020) proposed methods that do not assume any word alignment or dictionary supervision at training time to handle unseen terms at test time. For their flexibility, we choose them as our baselines. As discussed in Section 1, most previous methods are trained and evaluated on general domain corpora. In this work, we instead tackle highly specialized domain-specific corpora such as law and medicine, where the terms are much longer and often rare.

Domain-specific NMT Another line of research related to our problem is domain-specific NMT, where difficulties arise from both a limited amount of parallel data and specialized lexicons. Similar to the hard approaches in lexically constrained NMT, several works rely on domain-specific dictionaries (Zhang and Zong, 2016a; Hu et al., 2019; Thompson et al., 2019; Peng et al., 2020) when generating translations, but they are also prone to the same issues. Other domain-specific NMT methods include unsupervised lexicon adaptation (Hu et al., 2019), synthetic parallel data generation with monolingual data (Sennrich et al., 2016a), and multi-task learning that combines language modeling and translation objectives (Gulcehre et al., 2015; Zhang and Zong, 2016b; Domhan and Hieber, 2017). Our method is a form of multi-task learning by utilizing both the source and target language text for an additional task, while the previous works mostly use only the target language text.

Dinu et al. (2019)'s dataset	
Iate-414	gold(15), CDU(13), bridge(12), China(11), Syria(11), night(11), campaign(11), generation(9), month(7), Iraq(7)
Iate-581	gold(26), doping(23), CDU(19), sport(17), US(15), bridge(14), Syria(13), campaign(13), China(11), night(11)
Wikt-727	percent(61), police(50), Thursday(41), Putin(19), five(17), September(14), Venus(13), later(12), Tuesday(11), less(11)
Wikt-975	percent(61), police(59), Thursday(44), Putin(24), old(21), September(21), five(16), swimmer(14), later(13), Venus(13)

Our dataset	
Law (De-En)	Council(706), Regulation(521), Commission(481), Union(478), Treaty(345), Official Journal(319), Member State(283), proposal(239), on a proposal from the Commission(229), market(181)
Medical (De-En)	injection(469), water(275), water for injection(270), patient(269), infusion(258), solution for infusion(226), sodium(159), distribution(127), volume of distribution(125), treatment(120)
Law (Ko-En)	si(451), official(445), public official(436), member(436), term of office(367), gu(265), education(209), period(180), term(180), management(156)

Table 1: Top 10 most frequent terms in Dinu et al. (2019)'s and our test splits. Numbers in parenthesis indicate the frequency of terms in each data. As shown in the two tables, all top 10 terms in the WMT corpus are unigrams, while there are longer terms (up to 6-grams) in the domain-specific corpora. Furthermore, compared to WMT, the terms in the domain-specific corpora are more specialized for their corresponding domains.

Span-based Masking Span-based masking is to predict the spans of masked tokens, as opposed to individual token predictions in BERT (Devlin et al., 2019). With this training objective, the model showed improved performance on span-level tasks including question answering and coreference resolution (Joshi et al., 2019). Concurrently, autoregressive sequence-to-sequence pre-trained models also utilized span-based masking as their objectives and demonstrated its effectiveness in many downstream tasks (Song et al., 2019b; Lewis et al., 2020; Raffel et al., 2020). Similar to theirs, our training scheme takes advantage of autoregressive

span-based prediction but we condition on both the source language and the previous non-masked target language tokens.

3 Approach

3.1 Source-Conditioned Masked Span Prediction

We posit that adopting auxiliary span-level supervision in generation can benefit both short and long terminology and sentence-level translation. We, therefore, propose an extra span-level prediction task in translation—namely, source-conditioned masked span prediction (SSP). Different from the recent sequence-to-sequence pre-trained models (Song et al., 2019b; Lewis et al., 2020; Raffel et al., 2020), our approach applies span masking only on the target side. By conditioning on the full context of the source language and the previous non-masked target language tokens (due to autoregressive decoding), the model is forced to predict the spans of missing tokens given fully referenced information in the encoder and partially in the decoder.

Span masking We follow the masking procedure proposed in SpanBERT (Joshi et al., 2019), where we first sample the length of spans from a clamped geometric distribution (p=0.2, max=10) and then corrupt 80% of masked tokens with [MASK], 10% with random tokens, and 10% unchanged. We set the corruption ratio to 50%.

Multi-task Learning As our training scheme consists of two objectives (*i.e.* translation and masked span prediction), we define the total training objective as follows. Let θ be the model parameter and $\mathbb{C}$ be the term-matched corpus where each sentence contains at least one or more terms. The first objective, translation, is to maximize the likelihood of the conditional probability of y:

$$p_\theta(y|x) = \prod_{t=1}^{T+1} p_\theta(y_t|y_{0:t-1}, x), \qquad (1)$$

where $y = (y_1, \ldots, y_T)$ is the target ground-truth (GT) sentence with length T and $x = (x_1, \ldots, x_S)$ is the source sentence with length S. For the SSP objective, we first corrupt random spans of y until the corruption ratio, resulting in $\tilde{y}$. Then we autoregressively predict the masked tokens $\bar{y}$ while conditioned on both $\tilde{y}$ and x:

$$p_\theta(\bar{y}|\tilde{y}, x) = \prod_{t=1}^{T+1} m_t p_\theta(y_t|\tilde{y}_{0:t-1}, x), \qquad (2)$$

where $m_t = 1$ means y_t is masked.

Finally, we simultaneously optimize the joint loss:

$$\mathcal{L}_{total} = -\frac{1}{|\mathbb{C}|} \sum_{\substack{(x,y)\sim\mathbb{C}, \\ \tilde{y}\sim C(y)}} \log p_\theta(y|x) + \gamma \log p_\theta(\bar{y}|\tilde{y}, x), \qquad (3)$$

where C is a span-level corrupter and γ is a task coefficient that weights the relative contribution of SSP.

		# Sent.	Avg. words per sent.	# Terms	Avg. terms per sent.	# Unique terms
Law (De→En)	SRC	447,410 (441K/3K/3K)	27.46	1,677,852	2.33	25,460
	TRG		30.77			27,755
Medical (De→En)	SRC	494,316 (488K/3K/3K)	19.01	1,494,269	1.34	8,633
	TRG		20.25			8,990
Law (Ko→En)	SRC	93,240 (87K/3K/3K)	16.52	353,894	3.52	2,354
	TRG		34.56			2,733

Table 2: Statistics of the filtered corpus and matched terms. Note that *# unique terms* in the source (SRC) and target (TRG) languages are not the same. For instance, "Arzneimittel" can translate into multiple forms—"pharmaceutical products", "drug", "medicinal product", etc.— depending on the context.

4 Experiments

4.1 Setup

Data We use De-En legal and medical domain corpora from OPUS[1] (Tiedemann, 2012) and the De-En bilingual term dictionary from IATE[2]. Terms in different languages are aligned via term IDs. For the typologically distant pair of languages, we use the Ko-En legal domain corpus available on AI Hub[3], and the manually processed bilingual term dictionary downloaded from the Korea Legislation Research Institute (KLRI) website[4]. In cases where

[1] http://opus.nlpl.eu/
[2] https://iate.europa.eu
[3] https://www.aihub.or.kr/aidata/87
[4] https://www.moleg.go.kr/board.es?mid=a10504000000&bid=0010&act=view&list_no=43927&nPage=2

Model	Law (De→En)							Law (Ko→En)						
	BLEU	Term% (↑)				LSM-3 (↑)		BLEU	Term% (↑)				LSM-3 (↑)	
		1-gram	2-gram	2>micro	2>macro	2>micro	2>macro		1-gram	2-gram	2>mirco	2>marco	2>micro	2>macro
GU19	70.64	94.36	92.33	73.31	45.25	86.22	74.92	51.31	81.47	76.51	58.15	38.51	69.41	62.52
VASWANI17	75.24	95.80	93.87	80.29	55.31	89.71	79.77	53.01	84.97	81.29	65.79	54.55	74.29	70.56
+SSP	**75.44**	**96.01**	94.08	**81.52**	**58.79**	**90.81**	**82.50**	**53.80**	**85.84**	**83.94**	**66.84**	**58.15**	75.71	69.82
CHEN20	74.19	95.55	94.08	80.63	54.73	89.80	80.89	53.08	85.49	83.25	65.51	52.49	74.53	67.81
+SSP	75.24	95.92	**94.50**	81.31	56.33	90.40	81.91	53.32	85.63	82.10	66.19	56.50	**76.02**	**72.27**

Table 3: (*Without dictionary*) Results on legal domain corpora (De→En and Ko→En) without terminology guidance at test time. VASWANI17 combined with our training objective (Eq.(3)) outperforms other methods in most cases. Note that GU19 is a non-autoregressive model, therefore not applicable to our proposed method. Higher Term% and LSM-3 mean better performance.

Model	Law (De→En)							Law (Ko→En)						
	BLEU	Term% (↑)				LSM-3 (↑)		BLEU	Term% (↑)				LSM-3 (↑)	
		1-gram	2-gram	2>micro	2>macro	2>micro	2>macro		1-gram	2-gram	2>mirco	2>marco	2>micro	2>macro
SUSANTO20	62.20	94.38	92.95	82.06	**64.06**	**94.93**	**92.14**	50.56	81.67	76.74	58.47	38.66	69.63	62.60
CHEN20	73.05	96.64	93.29	78.73	51.47	90.00	80.29	52.60	84.74	83.94	67.33	59.53	75.59	74.54
+SSP	**74.72**	**97.15**	**95.95**	**84.67**	57.48	93.94	83.62	**53.38**	**95.86**	**94.92**	**88.58**	**79.34**	**94.17**	**91.48**

Table 4: (*With dictionary*) Results on legal domain corpora when the GT terms are provided at test time. +SSP consistently shows improvements over its MLE counterparts. Contrary to the previous findings (Susanto et al., 2020; Chen et al., 2020), the models do not show improved BLEU scores compared to those in Table 3. We argue that providing terms at test time is indeed helpful for terminology generation, but it can often hinder the generation of fluent text. This becomes more apparent in our non-autoregressive setup.

one term translates into multiple terms, we consider all possible pairs to maximize the number of sentence and term matches.

To avoid trivial matches between the parallel sentences and terms, we filtered out terms that are less than four characters long and longer than 20 grams. Sentences that do not contain any term are also removed. The statistics of the datasets are reported in Table 2. More details about the preprocessing steps are in Appendix A.1.

For data splitting, we developed a new data splitting algorithm that considers the same distribution of n-grams across each data split. We use 3,000 sentences for valid and test sets in case of high redundancy in certain corpora, while previous works that utilize OPUS use only 2,000 (Koehn and Knowles, 2017; Müller et al., 2020). It is important to note that all the sentences in our data splits are matched with domain-specific terms (i.e. at least one or more terms exist in each sentence) following the style of Dinu et al. (2019). The pseudo-code for the terminology-aware data split algorithm is in Appendix B.

Baselines We compare our method on two recent lexically constrained NMT models of different natures: autoregressive (Chen et al., 2020) and non-autoregressive (Susanto et al., 2020), but both can operate with or without a term dictionary at test time. We refer to them as CHEN20 and SUSANTO20, respectively. +SSP indicates models trained with our proposed training scheme, while no indication is the standard MLE method. A base Transformer (Vaswani et al., 2017), denoted as VASWANI17, and a Levenshtein Transformer (Gu et al., 2019), denoted as GU19, are also reported to compare the relative performance between models. SUSANTO20 without a dictionary is equivalent to GU19.

Evaluation We use SacreBLEU[5] (Post, 2018) for measuring translation quality. For terminology translation, we use *term usage rate* for both short (≤2-grams) and long (>2-grams) terms. Term usage rate (Term%) is the number of generated terms divided by the total number of terms (Dinu et al., 2019; Susanto et al., 2020). Specifically for evaluating long terms, we report both the macro and micro averages due to the heavy-tailed nature of n-grams. In addition, although exact term translation is the primary objective for terminology translation, due to its harshness, evaluating models only with Term% may not fully describe the models'

[5] https://github.com/mjpost/sacrebleu

Model	Medical (De→En)						
	BLEU	Term% (↑)				LSM-3 (↑)	
		1-gram	2-gram	2>micro	2>macro	2>micro	2>macro
Without dictionary							
GU19	70.85	93.83	91.24	77.46	53.66	86.15	75.21
VASWANI17	76.31	94.22	90.80	79.82	61.03	87.11	80.48
+SSP	**76.87**	94.36	**91.31**	**80.63**	53.68	88.01	74.76
CHEN20	74.84	94.29	90.61	79.42	**68.37**	87.13	**84.64**
+SSP	76.72	**94.61**	90.42	80.41	68.03	**88.08**	83.04
With dictionary							
SUSANTO20	62.20	91.01	92.64	88.09	67.22	**95.46**	**94.27**
CHEN20	72.84	94.40	93.58	83.77	67.98	89.95	86.70
+SSP	**75.50**	**95.86**	**94.92**	**88.58**	**79.34**	94.17	91.48

Table 5: Results on the medical domain dataset (De→En).

behavior. Therefore, we also evaluate each model in terms of partial n-gram matches, which is explained in the next paragraph. All evaluations are conducted with a beam size of 5.

Partial N-gram Match Inspired by the longest common substring problem (Gusfield, 1997), we devised a partial n-gram match score for evaluating long terminology—*longest sub n-gram match* (LSM) score. Formally, let the generated target sentence be $\hat{y} = (\hat{y}_1, ..., \hat{y}_T)$ and the matched terms for the target ground truth (GT) sentence y be $y' = \bigcup_{i=1}^{N}(y'_{i1}, ..., y'_{il})$, where N is the number of GT terms in y and l is an arbitrary n-gram length for i-th term. Then, LSM is defined as the ratio of the longest n-gram overlap divided by l. As too many overlaps can occur at the uni and bi-gram levels, we only calculate LSM for long terminology, which means the least overlap has to be greater than or equal to 3 grams, all else being zero, therefore denoted as LSM-3.

4.2 Results and Analysis

For the legal domain, where many terms are exceptionally long compared to most other domains, our training scheme shows consistent improvements over the standard MLE counterparts, as shown in Table 3 and Table 4. Even with the extreme setting of law Ko→En, low-resourced and typologically divergent, our method is still effective in most metrics we use. Compared to the autoregressive models, GU19 and SUSANTO20 did not achieve competitive BLEU scores in our domain-specific setup. We suspect that this is due to both its complex decoding nature and the small amount of training data (originally WMT). Sampled translation results are reported in Table 9.

For the medical domain, the behaviors of two baselines, VASWANI17 and CHEN20, are not clearly shown compared to the legal domain. However, our training scheme shows consistent improvements in BLEU and Term% at 2>micro which reflects the global performance of long terminology generation. Similar to the legal De→En results, SUSANTO20 shows better performance on several metrics on long terminology translation, but the BLEU score is decreased by about 8 points, compared to no dictionary use.

5 Conclusion

We propose a simple and effective training scheme for improving lexically constrained NMT by introducing the masked span prediction task on the decoder side. Our method shows its effectiveness in terms of terminology and sentence-level translation over the standard MLE training in highly specialized domains in two language pairs. As we publicly release our code and datasets, we hope that more people can join this area of research without much burden. In the future, we plan to further investigate applying our method to non-autoregressive methods.

Acknowledgments

We want to thank Minjoon Seo for his constructive comments on our research direction and Wooju Kim for his help in starting this topic. We also thank anonymous reviewers for their effort and valuable feedback. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), and National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945), funded by the Korea government (MSIT).

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof

Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark. Association for Computational Linguistics.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization. Main track.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Miriam Exel, Bianka Buschbeck, Lauritz Brandt, and Simona Doneva. 2020. Terminology-constrained neural machine translation at SAP. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 271–280, Lisboa, Portugal. European Association for Machine Translation.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Dan Gusfield. 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open

source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Eunjeong L Park and Sungzoon Cho. 2014. Konlpy: Korean natural language processing in python. In *Annual Conference on Human and Language Technology*, pages 133–136. Human and Language Technology.

Wei Peng, Chongxuan Huang, Tianhao Li, Yun Chen, and Qun Liu. 2020. Dictionary-based data augmentation for cross-domain neural machine translation. *arXiv preprint arXiv:2004.02577*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8886–8893.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019a. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019b. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Brian Thompson, Rebecca Knowles, Xuan Zhang, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. HABLex: Human annotated bilingual lexicons for experiments in machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1382–1387, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jiajun Zhang and Chengqing Zong. 2016a. Bridging neural machine translation and bilingual dictionaries. *arXiv preprint arXiv:1610.07272*.

Jiajun Zhang and Chengqing Zong. 2016b. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

A Preprocessing and Training

A.1 Preprocessing

For De→En, we applied Moses tokenization (Koehn et al., 2007) and joint source-target byte pair encoding (BPE) (Sennrich et al., 2016b) with 20,000 split operations. For En→Ko, English was tokenized using spaCy[6] and Korean using KoNLPy's MeCab-ko[7] (Park and Cho, 2014), followed by BPE with 20,000 operations. We apply sentence filtering up to 80 tokens.

A.2 Training

Model We follow the base Transformer architecture and fix the same hyperparameter configurations for all baselines. For the exact implementation of each baseline, we followed the authors' official code on github (CHEN20[8] and SUSANTO20[9]). We implemented our code using FAIRSEQ[10] (Ott et al., 2019), trained on Nvidia GeForce RTX 3090 and RTX 2080 Ti GPUs.

Hyperparameter Detailed hyperparameter settings of baselines are reported below. Span masking and task coefficient only apply to our proposed training scheme.

Transformer	
Embedding dim.	512
Transformer FFN dim.	2048
Enc/Decoder layers	6
Attention heads	8
Share all embedding	True
Dropout	0.3
Label smoothing	0.1
Optimizer	Adam
Learning rate	0.0005
Warmup updates	4000
Maximum token per batch	4096
Maximum token lengths	80
Span Masking	
Span length	Geometric (p=0.2)
Maximum span length	10
Minimum span length	1
Corruption ratio	0.5
Task Coefficient	
Task coefficient (γ)	0.5

Table 6: Hyperparameter settings

[6] https://spacy.io/

[7] https://konlpy.org/en/latest/

[8] https://github.com/ghchen18/leca

[9] https://github.com/raymondhs/constrained-levt

[10] https://github.com/pytorch/fairseq

B Terminology-Aware Data Split Algorithm

Algorithm 1: Terminology-Aware Data Split Algorithm

Data: Dictionary $\mathbb{D}$, Corpus $\mathbb{C}$, Held-out Size R
Result: Sent=(Sent$_{train}$, Sent$_{valid}$, Sent$_{test}$)
Term=(Term$_{train}$, Term$_{valid}$, Term$_{test}$)

1: Sort $\mathbb{D}$ in a descending order
2: N = dict()
3: T' = dict()
4: S = ($|\mathbb{C}| - 2 * R, R, R$)
5: Sent$_{train}$ = [], Sent$_{valid}$ = [], Sent$_{test}$ = []
6: Term$_{train}$ = [], Term$_{valid}$ = [], Term$_{test}$ = []
7: **for** i in {1,2, ..., $|\mathbb{C}|$} **do**
8: x, y = $\mathbb{C}[i]$
9: *ngramlist* = []
10: T'' = []
11: **for** (x', y') in $\mathbb{D}$ **do**
12: **if** y' in y and x' in x **then**
13: *ngramlist*.append(ngram(y'))
14: y = y.replace(y', "", 1)
15: x = x.replace(x', "", 1)
16: T''.append((x', y'))
17: **end if**
18: **end for**
19: n = Max(*ngramlist*)
20: **if** n is not in N.keys() **then**
21: $N[n]$ = []
22: **end if**
23: $N[n]$.append(i)
24: $T'[i] = T''$
25: **end for**
26: K = Sort the keys in N in a descending order.
27: **for** k in K **do**
28: i_{dk} , i_{uk} = DuplicateCheck($N[k]$)
29: (Sent, Term) += Distributor_Dup(i_{dk}, $N[k]$, T', S)
30: (Sent, Term) += Distributor_Uni(i_{uk}, $N[k]$, T', S)
31: **end for**

Line 1 : Sort $\mathbb{D}$ w.r.t. target language terms.

Line 2 : Initialize a dictionary for storing paired sentences. The keys are the longest n-gram lengths for each sentence w.r.t. the target language.

Line 3 : Initialize a dictionary for storing matched terms. The keys are the indices of a corresponding sentence.

Line 13 : ngram() returns the token length of a term. In our case, it is used for calculating the length of a target language token y'.

Line 14 : Replace y' with "" in y to avoid unwanted substring duplication (e.g., In case of having "public officer" and "officer" in a sentence, we would like to first match "public officer" instead of "office" when we have "public officer" in the dictionary. See Line 1).

Line 19 : Calculate the maximum length of n-grams in y.

Line 23 : Store the sentence index w.r.t. its longest length of n-grams.

Line 24 : Store the list of terms w.r.t. its sentence index.

Line 28 : DuplicateCheck() checks for duplication in the corpus and returns duplicate and non-duplicate indices. Note

that i_{dk} is a list of duplicate sentence indices, and S_{uk} is a list of unique sentence indices.

Line 29 : Distributor_Dup() first calculates the number of sentences and phrases to be distributed across train, valid, and test sets following the ratio in *S*, and then distributes sentences accordingly.

Line 30 : Distributor_Uni() distributes unique sentences and phrases alternatively between train, valid, and test sets.

C Removing duplicates

As the OPUS datasets contain duplicate sentences (Aharoni and Goldberg, 2020), we further evaluate each model with unseen, unique test samples only. Similar to Tables 7 and 8, our training scheme outperforms its MLE counterparts. The Ko-En law corpus does not contain any duplicate sentence, and therefore the results are equivalent to those in Tables 3 and 4.

Model	Law (De→En)						
	BLEU	Term% (↑)				LSM-3 (↑)	
		1-gram	2-gram	2>micro	2>macro	2>micro	2>macro
Without dictionary							
GU19	68.14	93.71	91.87	72.32	46.24	85.05	74.22
VASWANI17	72.86	95.30	93.36	78.68	54.99	88.40	78.69
+SSP	**73.15**	**95.57**	93.54	**79.98**	**59.18**	**89.64**	**81.53**
CHEN20	71.89	95.04	93.54	78.83	54.55	88.43	79.63
+SSP	72.93	95.46	**94.14**	79.74	56.63	89.19	80.88
With dictionary							
SUSANTO20	59.23	94.10	92.89	82.86	**68.22**	**95.28**	**93.36**
CHEN20	70.90	96.36	94.01	77.26	51.97	89.12	80.06
+SSP	**72.70**	**96.85**	**95.68**	**83.65**	58.33	93.30	83.40

Table 7: Results on the law domain dataset with no duplication in data (De→En).

Model	Medical (De→En)						
	BLEU	Term% (↑)				LSM-3 (↑)	
		1-gram	2-gram	2>micro	2>macro	2>micro	2>macro
Without dictionary							
GU19	54.27	89.93	84.35	67.83	50.44	78.18	70.33
VASWANI17	58.29	90.09	84.51	70.98	57.82	79.18	76.45
+SSP	59.19	90.43	**85.50**	**71.51**	49.20	**80.38**	70.10
CHEN20	58.27	90.30	84.02	70.38	64.02	79.28	**80.14**
+SSP	**59.49**	**90.57**	83.53	71.25	**64.36**	80.29	78.70
With dictionary							
SUSANTO20	45.60	88.77	89.95	**86.86**	68.22	**94.83**	**93.86**
CHEN20	58.30	90.81	89.79	79.83	67.62	86.59	85.05
+SSP	**60.30**	**93.10**	**91.10**	85.19	**79.26**	91.34	90.51

Table 8: Results on the medical domain dataset with no duplication in data (De→En).

D Examples

Table 9 shows translation results of the baselines and our method.

Source	dieses Vorbringen wurde zurückgewiesen , da die einschlägigen Bestimmungen der Grundverordnung sehr wohl mit dem WTO-**Übereinkommen zur Durchführung des Artikels VI des Allgemeinen Zoll- und Handelsabkommens 1994** und dem **Übereinkommen über Subventionen und Ausgleichsmaßnahmen** vereinbar sind .
VASWANI17	this claim was rejected because the relevant provisions of the basic Regulation are very compatible with the 1994 WTO Agreement on Implementation of Article VI of the General Agreement on Tariffs and Trade and the 1994 **Agreement on Subsidies and Countervailing Measures** .
+SSP	this claim was rejected as the relevant provisions of the basic Regulation are indeed consistent with the WTO **Agreement on Implementation of Article VI of the General Agreement on Tariffs and Trade 1994** and the **Agreement on Subsidies and Countervailing Measures** .
CHEN20	this claim was rejected as the relevant provisions of the basic Regulation are , however , in any event , compatible with the WTO Agreement on the implementation of Article VI of the General Agreement on Tariffs and Trade 1994 and with the **Agreement on Subsidies and Countervailing Measures** .
+SSP	this claim was rejected because it is true that the relevant provisions of the basic Regulation are consistent with the **Agreement on Implementation of Article VI of the General Agreement on Tariffs and Trade 1994** and the **Agreement on Subsidies and Countervailing Measures** .
Reference	this claim was rejected on the grounds that the anti-circumvention provisions of the basic Regulation are not incompatible with the **Agreement on Implementation of Article VI of the General Agreement on Tariffs and Trade 1994** and the **Agreement on Subsidies and Countervailing Measures** .
Terminology	{**Übereinkommen zur Durchführung des Artikels VI des Allgemeinen Zoll- und Handelsabkommens 1994**→**Agreement on Implementation of Article VI of the General Agreement on Tariffs and Trade 1994**, **Übereinkommen über Subventionen und Ausgleichsmaßnahmen**→**Agreement on Subsidies and Countervailing Measure**s}

Source	(3) Das **Angebot zur vorzugsweisen Zeichnung** sowie die Frist , innerhalb deren dieses Recht ausgeuebt werden muß, sind Gegenstand einer Bekanntmachung in dem gemäß der Richtlinie 68 / 151 / EWG bestimmten einzelstaatlichen **Amtsblatt .**
VASWANI17	3 . the tender for subscription and the time limit within which that right must be exercised shall be published in the **national gazette** determined in accordance with Directive 68 / 151 / EEC .
+SSP	3 . the tender for a preference call and the time limit within which that right must be exercised shall be the subject of a notice in the **national gazette** designated in accordance with Directive 68 / 151 / EEC .
CHEN20	3 . the tender for preferred subscription and the time limit within which it must be exercised shall be the subject of a notice published in the national publication designated pursuant to Directive 68 / 151 / EEC .
+SSP	3 . tenders for preference drawing together with the time limit within which that right must be exercised shall be the subject of a notice in the **national gazette** designated in accordance with Directive 68 / 151 / EEC .
Reference	any **offer of subscription on a pre-emptive basis** and the period within which this right must be exercised shall be published in the **national gazette** appointed in accordance with Directive 68 / 151 / EEC .
Terminology	{**Angebot zur vorzugsweisen Zeichnung**→**offer of subscription on a pre-emptive basis**, **Amtsblatt**→**national gazette**}

Source	(19) Nach der **Rechtsprechung** des **Gerichtshofs** sind einzelstaatliche Vorschriften betreffend die Fristen für die Rechtsverfolgung **zulässig** , sofern sie für derartige Klagen nicht ungünstiger sind als für gleichartige Klagen , die das innerstaatliche Recht betreffen , und sofern sie die Ausübung der durch das **Gemeinschaftsrecht** gewährten Rechte nicht praktisch unmöglich machen .
VASWANI17	(19) According to the case law of the **Court of Justice** , national rules concerning time limits for bringing actions may be allowed , provided that such actions are not less favourable than those relating to the like actions under national law and do not make it impossible in practice to exercise the rights conferred by **Community law** .
+SSP	(19) According to the case law of the **Court of Justice** , national rules concerning the time limits for prosecution are **admissible** , provided that they are not less favourable to such actions than those for similar actions under national law if they do not make it impossible to exercise the rights conferred by **Community law** in practice .
CHEN20	(19) According to the case law of the **Court of Justice** , national rules on the time limits for the exercise of jurisdiction may be allowed , provided that they are not less favourable for such actions than for the same actions covered by national law and do not make it impossible in practice to exercise the rights conferred by **Community law** .
+SSP	(19) The **Court of Justice** has **case-law** that national provisions relating to time limits for bringing actions may be accepted , provided that they are no less favourable in such actions than those relating to similar actions brought under national law , provided that they do not practically make it impossible for the exercise of rights conferred by **Community law** .
Reference	(19) According to the **case-law** of the **Court of Justice** , national rules relating to time limits for bringing actions are **admissible** provided that they are not less favourable than time limits for similar actions of a domestic nature and that they do not render the exercise of rights conferred by the **Community law** impossible in practice .
Terminology	{**Gerichtshof**→**Court of Justice**, **Gemeinschaftsrecht**→**Community law**, **zulässig**→**admissible**, **Rechtsprechung**→**case-law**}

Table 9: Translation outputs of the models trained with or without our method.

Quotation Recommendation and Interpretation Based on Transformation from Queries to Quotations

Lingzhi Wang[1,2], **Xingshan Zeng**[3], **Kam-Fai Wong**[1,2]
[1]The Chinese University of Hong Kong, Hong Kong, China
[2]MoE Key Laboratory of High Confidence Software Technologies, China
[3]Huawei Noah's Ark Lab, China
[1,2]{lzwang,kfwong}@se.cuhk.edu.hk [3]zeng.xingshan@huawei.com

Abstract

To help individuals express themselves better, quotation recommendation is receiving growing attention. Nevertheless, most prior efforts focus on modeling quotations and queries separately and ignore the relationship between the quotations and the queries. In this work, we introduce a transformation matrix that directly maps the query representations to quotation representations. To better learn the mapping relationship, we employ a mapping loss that minimizes the distance of two semantic spaces (one for quotation and another for mapped-query). Furthermore, we explore using the words in history queries to interpret the figurative language of quotations, where quotation-aware attention is applied on top of history queries to highlight the indicator words. Experiments on two datasets in English and Chinese show that our model outperforms previous state-of-the-art models.

1 Introduction

Quotations are essential for successful persuasion and explanation in interpersonal communication. However, it is a daunting task for many individuals to write down a suitable quotation in a short time. This results in a pressing need to develop a quotation recommendation tool to meet such a demand.

To that end, extensive efforts have been made to **quotation recommendation**, which aims to recommend an ongoing conversation with a quotation whose sense continues with the existing context (Wang et al., 2020). As quotations are concise phrases or sentences to spread wisdom, which are always in figurative language and difficult to understand, they are assumed written in a different pseudo-language (Liu et al., 2019a). Intuitively, we can infer the meanings of quotations by their neighborhood contexts, especially by the *query* turn (the last turn of conversation that needs recommendation).

$[t_1]$: Save your money. Scuf is the biggest ripoff in gaming. $[t_2]$: What would you suggest instead? $[t_3]$: Just use a normal controller. $[t_4]$: Oooooh, I get it now...you're just dumb. $[t_5]$: The dumb ones are the people spending over $100 for a controller. [*A fool and his money are soon parted.*]
$[h_1]$: Anyone that spends that much money just to get different writing on a box..... [*A fool ... parted.*] $[h_2]$: And that's probably why you'll never have a billion dollars. [*A fool ... parted.*] $[h_3]$: Seriously. Why do people not do market research before buying something!?! [*A fool ... parted.*]

Figure 1: A Reddit conversation snippet (upper part) with three history queries (lower part). Quotations to be recommended are in square brackets. Indicative words are on wavy-underline.

To illustrate our motivation, Figure 1 shows a Reddit conversation with some history queries associated with quotation Q, "*A fool and his money are soon parted*". From the queries (t_5 and h_1 to h_3), we can infer the meaning of quotation Q is "A foolish person spends money carelessly and won't have a lot of money." based on the contexts. From h_3, we can also know the implication behind the words, which is "Do a marketing research before buying". Humans can establish such a relationship between quotations and queries and then decide what to quote in their writings, so can machines (neural network). Therefore, we introduce a transformation matrix, in which machines can learn the direct mapping from queries to quotations. The matrix is worked on the outputs of two encoders, conversation encoder and quotation encoder, encoding conversation context and quotations respectively.

Furthermore, we can use the words in the queries to interpret quotations. h_1 to h_3 in Figure 1 are

The code is available at https://github.com/Lingzhi-WANG/Quotation-Recommendation

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 754–758
August 1–6, 2021. ©2021 Association for Computational Linguistics

denoted as *history queries*, and the words on wavy-underline are denoted as indicators to quotations. It can be seen that we can interpret quotations by highlighting the words in the queries. Therefore, we compute quotation-aware attention over all the history queries (after the same transformation as we mentioned before) and then display indicators we learned, which also reflects the effectiveness of the transformation.

In summary, we introduce a transformation between the query semantic space and quotation semantic space. To minimize the distance of their semantic space after transformation mapping, an auxiliary mapping loss is employed. Besides, we propose a way to interpret quotations with indicative words in the corresponding queries.

The remainder of this paper is organized as follows. The related work is surveyed in Section 2. Section 3 presents the proposed approach. And Section 4 and 5 present the experimental setup and results respectively. Finally, conclusions are drawn in Section 6.

2 Related Work

Quotation Recommendation. In previous works on quotation recommendation, some efforts are made for online conversations (Wang et al., 2020; Lee et al., 2016) and some for normal writing (Liu et al., 2019a; Tan et al., 2015, 2016). Our work focuses on the former. For methodology, the methods they applied can be divided into generation-based framework (Wang et al., 2020; Liu et al., 2019a) and ranking framework (Lee et al., 2016; Tan et al., 2015, 2016). Different from previous works which mainly focus on separate modeling of quotation and query and pay little attention to the relationship between them, our model directly learns the relationship between quotations and query turns based on a mapping mechanism. The relationship mapping is jointly trained with the quotation recommendation task, which improves the performance of our model.

3 Our model

This section describes our quotation recommendation model, whose overall structure is shown in Figure 2. The input of the model mainly contains the observed conversation c and the quotation list q. The conversation c is formalized as a sequence of turns (e.g., posts or comments) $\{t_1, t_2, ..., t_{n_c}\}$ where n_c represents the length of the conversation

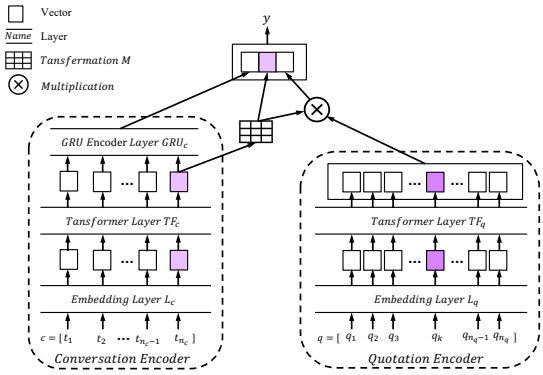


Figure 2: Our model for quotation recommendation.

(number of turns) and t_{n_c} is the query turn. t_i represents the i-th turn of the conversation and contains words $\boldsymbol{w}_i$. The quotation list q is $\{q_1, q_2, ..., q_{n_q}\}$, where n_q is the number of quotations and q_k is the k-th quotation in list q, containing words $\boldsymbol{w}'_k$. Our model will output a label $y \in \{1, 2, ..., n_q\}$, to indicate which quotation to recommend.

3.1 Conversation Modeling

Our model encodes the observed conversation c with a hierarchical structure, which is divided into three parts. The first part is an embedding layer mapping the words $\boldsymbol{w}_i$ in each turn t_i into vectors. We then apply transformer (Vaswani et al., 2017) to learn the representation for each turn. Similar to BERT (Devlin et al., 2018), we only use the encoder of transformer, which is stacked of several self-attention and feed-forward layers. We add a token [CLS] at the beginning of each turn. The hidden representation of [CLS] after transformer encoder is defined as the turn representation $\boldsymbol{r}_i^t$ of turn t_i. The procedures for the first two parts are summarized as follows:

$$\boldsymbol{h}_i^T = \mathrm{FFN}(\mathrm{Self\_Attention}(\mathrm{Embed}([w_0; \boldsymbol{w}_i]))) \quad (1)$$

where w_0 represents the [CLS] token, and $[;]$ indicates concatenation. Therefore $\boldsymbol{r}_i^t = \boldsymbol{h}_{i,0}^T$.

Next, we use a Bi-GRU (Cho et al., 2014) layer to model the whole conversation structure. With the turn representations $\{\boldsymbol{r}_1^t, \boldsymbol{r}_2^t, ..., \boldsymbol{r}_{n_c}^t\}$ ($\boldsymbol{r}_{n_c}^t$ is the representation for the query turn) of conversation c derived from previous procedure, the hidden states are updated as follows:

$$\overrightarrow{\boldsymbol{h}}_i^G = \overrightarrow{\mathrm{GRU}}(\overrightarrow{\boldsymbol{h}}_{i-1}^G, \boldsymbol{r}_i^t), \ \overleftarrow{\boldsymbol{h}}_i^G = \overleftarrow{\mathrm{GRU}}(\overrightarrow{\boldsymbol{h}}_{i+1}^G, \boldsymbol{r}_i^t) \quad (2)$$

Finally, we define the conversation representation as the concatenation of the final hidden states from two directions: $\boldsymbol{h}^c = [\overrightarrow{\boldsymbol{h}}_{n_c}^G; \overleftarrow{\boldsymbol{h}}_1^G]$.

3.2 Quotation Modeling

For each quotation q_k in list q, we extract quotation representation $\boldsymbol{r}_k^q$ with similar operation as turn representations (see Eq. 1). As Liu et al. (2019b) points out, the language used in quotations is usually different from our daily conversations, which results in two different semantic spaces. Therefore, we do not share the parameters of the embedding layer and transformer layers for quotations and conversation turns. We concatenate all the quotation representations and get a combined quotation matrix $\boldsymbol{Q}$, which includes n_q rows and each row represents one quotation.

3.3 Recommendation Based on Transformation

To perform a reasonable recommendation, we consider the observed conversation c, the query turn t_{n_c} as well as the quotation list q. Since they are in different semantic spaces (Section 3.2), we first map the query turns into the space of quotations with a transformation matrix $\boldsymbol{M}$. We assume with such transformation, the space gap can be resolved. Thus, we can calculate the distance between queries and quotations. We use $\boldsymbol{z}^c$ to represents the distances between $\boldsymbol{r}_{n_c}$ and the quotations, and it is defined with the following equation:

$$\boldsymbol{z}^c = \boldsymbol{Q} \times (\boldsymbol{M}\boldsymbol{r}_{n_c}) \tag{3}$$

Finally, the output layer is defined as:

$$\boldsymbol{y} = \boldsymbol{W}[\boldsymbol{z}^c; \boldsymbol{h}^c; \boldsymbol{M}\boldsymbol{r}_{n_c}] + \boldsymbol{b} \tag{4}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable parameters. We recommend the quotations with the top n highest probabilities, which are derived with a softmax function:

$$p(\hat{q} = i) = \frac{\exp(y_i)}{\sum_{k=1}^{n_q} \exp(y_k)} \tag{5}$$

3.4 Training Procedure

We define our training objective as two parts. The first part is called recommendation loss, which is the cross entropy over the whole training corpus $\mathbb{C}$:

$$\mathcal{L}_{rec} = -\sum_{c \in \mathbb{C}} \log\, p(\hat{q} = q^c | c, q) \tag{6}$$

where q^c is the ground-truth quotation for conversation c in training corpus. The second part is to help on the learning of transformation matrix $\boldsymbol{M}$, where we minimize the distance between the transformed query turn representation and the corresponding ground-truth quotation:

$$\mathcal{L}_{map} = \sum_{c \in \mathbb{C}} ||\boldsymbol{M}\boldsymbol{r}_{n_c} - \boldsymbol{r}_{q^c}^q||_2^2 \tag{7}$$

To train our model, the final objective is to minimize $\mathcal{L}$, the combination of the two losses:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \cdot \mathcal{L}_{map} \tag{8}$$

where λ are the coefficient determining the contribution of the latter loss.

4 Experimental Setup

Datasets. We conduct experiments based on datasets from two different platforms, Weibo and Reddit, released by Wang et al. (2020). To make our experimental results comparable to Wang et al. (2020), we utilize their preprocessed data directly.

Parameter Setting. We first initialize the embedding layer with 200-dimensional Glove embedding (Pennington et al., 2014) for Reddit and Chinese words embedding (Song et al., 2018) for Weibo. For transformer layers, we choose the number of layers and heads as (2, 3) for Reddit and (4, 4) for Weibo. For the hidden dimension of transformer layers and BiGRU layers (each direction), we set it to 200. We employ Adam optimizer (Kingma and Ba, 2015) with initial learning rate with 1e-4 and early stop adoption (Caruana et al., 2001) in training. The batch size is set to 32. Dropout strategy (Srivastava et al., 2014) and L_2 regularization are used to alleviate overfitting. And the tradeoff parameter λ is chosen from {1e-4, 1e-3}. All the hyper-parameters above are tuned on the validation set by grid search.

Evaluation and Comparisons. Our model returns a quotation list arranged in descending order of likelihood of recommendation for each conversation. Therefore, we adopt MAP (Mean Average Precision), P@1 (Precison@1), P@3 (Precison@3), and nDCG@5 (normalized Discounted Cumulative Gain@5) for evaluation.

For comparison, we compare with previous works that focus on quotation recommendation. Below shows the details:

1) LTR (Learning to Rank). We first collect features (e.g., frequency, Word2Vec, etc.) mentioned in Tan et al. (2015) and then use the learning to rank tool RankLib [1] to do the recommendation.

[1] https://github.com/danyaljj/rankLib1

Models	Weibo				Reddit			
	MAP	P@1	P@3	NG@5	MAP	P@1	P@3	NG@5
Baselines								
LTR	9.3	3.6	8.5	8.1	7.1	1.7	6.4	6.2
CNN-LSTM	11.3	7.3	11.0	10.8	5.2	4.1	7.0	6.9
NCIR	26.5	22.6	27.8	26.7	12.2	7.3	12.3	11.4
CTIQ	30.3	27.2	33.2	31.6	21.9	17.5	25.8	23.8
BERT	31.4	27.9	34.0	32.3	26.4	18.0	30.2	28.5
OUR MODEL	**34.9**	**30.3**	**36.1**	**34.9**	**31.8**	**23.3**	**35.0**	**32.1**

Table 1: Main comparison results on Weibo and Reddit datasets (in %). NG@5 refers to NDCG@5. The best results in each column are in **bold**. Our model yields significantly better scores than all other comparisons for all metrics ($p < 0.01$, paired t-test).

2) CNN-LSTM. We implement the model proposed in Lee et al. (2016), which adopts CNN to learn the semantic representation of each turn and then uses LSTM to encode the conversation.
3) NCIR. It formulates quotation recommendation as a context-to-quote machine translation problem by using the encoder–decoder framework with attention mechanism (Liu et al., 2019b).
4) CTIQ. The SOTA model (Wang et al., 2020), which employs an encoder-decoder framework enhanced by Neural Topic Model to continue the context with a quotation via language generation.
5) BERT. We encode the conversation by BiLSTM on the BERT representations for the turns, followed by a prediction layer.

5 Experimental Results

5.1 Quotation Recommendation

Table 1 displays the recommendation results comparing our model with the baselines on Weibo and Reddit datasets. Our model achieves the best performance, exceeding the baselines by a large margin, especially on Reddit dataset. The fact that better performance comes from BERT and our model indicates the importance of learning efficient content representations. Our model further considers the mapping between different semantic spaces, resulting in the best performance.

Ablation Study. We conduct an ablation study to examine the contributions of different modules in our model. We replace the transformer layers with Bi-GRU (W/O Transformer) to examine the effects of different turn encoders. We also compare the models by removing transformation matrix M (W/O M) or mapping loss $\mathcal{L}_{map}$ (W/O $\mathcal{L}_{map}$). The results are shown in Table 2. As can be seen, each module in our model plays a role in improving performance. The largest improvement comes from applying transformers as our encoders. The performance drop due to removing transformation and mapping loss justifies our assumption of different semantic spaces between quotations and queries.

Models	Weibo			Reddit		
	MAP	P@1	NDCG@5	MAP	P@1	NDCG@5
W/O Transformer	29.9	25.9	29.8	25.8	17.4	25.7
W/O M	31.7	27.4	31.8	29.5	21.6	29.5
W/O $\mathcal{L}_{map}$	32.6	28.4	32.4	30.4	22.6	30.6
Full Model	34.9	30.3	34.9	31.8	23.3	32.1

Table 2: Comparison results of different variants of our model on Weibo and Reddit datasets (in %).

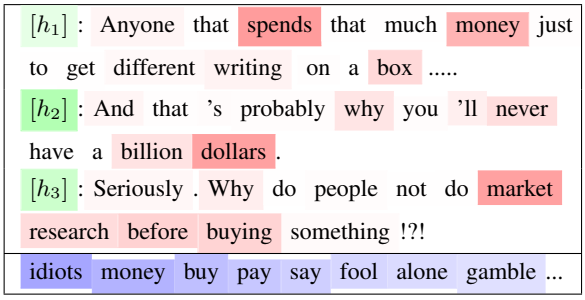


Figure 3: Upper part: example queries associated with the quotation "A fool and his money are soon parted.". Lower part: top 8 indicative words with the highest weighted summed self-attention scores. Darker colors represent higher weights.

5.2 Quotation Interpretation

We also explore how to interpret the figurative language of quotations with our model. We first extract the queries that are related to one certain quotation as history queries, then compute quotation-aware attention over all history queries. Specifically, for quotation q_k, with its relative history queries $\{h_1, h_2, ..., h_{m_k}\}$ from the corpus (m_k is the history number), we can compute their quotation-aware attention (query-level) with their representations derived from our model :

$$a_{k,i} = \frac{\exp(\boldsymbol{r}_k^q \cdot \boldsymbol{r}_{h_i})}{\sum_{j=1}^{m_k} \exp(\boldsymbol{r}_k^q \cdot \boldsymbol{r}_{h_j})} \quad (9)$$

On the other hand, we can extract the scores for the words in each history query with their self-attention weights (word-level) in transformer. Finally, the indicative words of one quotation are those with the highest scores after the multiplication of query-level and word-level attention scores.

Figure 3 shows an interpretation example. We display three example queries mentioned in Figure 1, with both their query-level attention (green) and

word-level attention (red). We can find that words like "spends", "money" and "dollars" are assigned higher scores since they are more related to the quotation topics . We also present the most indicative words derived from all history queries (the lower part of Figure 3). We can easily infer the meaning of the quotation with the help of indicative words like "idiots" and "buy".

6 Conclusion

In this paper, we propose a transformation from queries to quotations to enhance a quotation recommendation model for conversations. Experiments on Weibo and Reddit datasets show the effectiveness of our model with transformation. We further explore using indicative words in history queries to interpret quotations, which shows rationality of our method.

Acknowledgements

The research described in this paper is partially supported by HK GRF #14204118 and HK RSFS #3133237. We thank the three anonymous reviewers for the insightful suggestions on various aspects of this work.

References

Rich Caruana, Steve Lawrence, and C Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 957–960, New York, NY, USA. ACM.

Yuanchao Liu, Bo Pang, and Bingquan Liu. 2019a. Neural-based Chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yuanchao Liu, Bo Pang, and Bingquan Liu. 2019b. Neural-based Chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5522–5526, Florence, Italy. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 175–180.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015. Learning to recommend quotes for writing. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lingzhi Wang, Jing Li, Xingshan Zeng, Haisong Zhang, and Kam-Fai Wong. 2020. Continuity of topic, interaction, and query: Learning to quote in online conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6640–6650, Online. Association for Computational Linguistics.

Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence

Federico Bianchi
Bocconi University
Via Sarfatti 25, 20136
Milan, Italy
f.bianchi@unibocconi.it

Silvia Terragni
University of Milan-Bicocca
Viale Sarca 336, 20126
Milan, Italy
s.terragni4@campus.unimib.it

Dirk Hovy
Bocconi University
Via Sarfatti 25, 20136
Milan, Italy
dirk.hovy@unibocconi.it

Abstract

Topic models extract groups of words from documents, whose interpretation as a topic hopefully allows for a better understanding of the data. However, the resulting word groups are often not *coherent*, making them harder to interpret. Recently, neural topic models have shown improvements in overall coherence. Concurrently, contextual embeddings have advanced the state of the art of neural models in general. In this paper, we combine contextualized representations with neural topic models. We find that our approach produces more meaningful and coherent topics than traditional bag-of-words topic models and recent neural models. Our results indicate that future improvements in language models will translate into better topic models.

1 Introduction

One of the crucial issues with topic models is the quality of the topics they discover. *Coherent* topics are easier to interpret and are considered more meaningful. E.g., a topic represented by the words "apple, pear, lemon, banana, kiwi" would be considered a meaningful topic on *FRUIT* and is more coherent than one defined by "apple, knife, lemon, banana, spoon." Coherence can be measured in numerous ways, from human evaluation via intrusion tests (Chang et al., 2009) to approximated scores (Lau et al., 2014; Röder et al., 2015).

However, most topic models still use Bag-of-Words (BoW) document representations as input. These representations, though, disregard the syntactic and semantic relationships among the words in a document, the two main linguistic avenues to coherent text. I.e., BoW models represent the input in an inherently incoherent manner.

Meanwhile, pre-trained language models are becoming ubiquitous in Natural Language Processing (NLP), precisely for their ability to capture and maintain sentential coherence. Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), the most prominent architecture in this category, allows us to extract pre-trained word and sentence representations. Their use as input has advanced state-of-the-art performance across many tasks. Consequently, BERT representations are used in a diverse set of NLP applications (Rogers et al., 2020; Nozza et al., 2020).

Various extensions of topic models incorporate several types of information (Xun et al., 2017; Zhao et al., 2017; Terragni et al., 2020a), use word relationships derived from external knowledge bases (Chen et al., 2013; Yang et al., 2015; Terragni et al., 2020b), or pre-trained word embeddings (Das et al., 2015; Dieng et al., 2020; Nguyen et al., 2015; Zhao et al., 2017). Even for neural topic models, there exists work on incorporating external knowledge, e.g., via word embeddings (Gupta et al., 2019, 2020; Dieng et al., 2020).

In this paper, we show that adding contextual information to neural topic models provides a **significant** increase in topic coherence. This effect is even more remarkable given that we cannot embed long documents due to the sentence length limit in recent pre-trained language models architectures.

Concretely, we extend Neural ProdLDA (Product-of-Experts LDA) (Srivastava and Sutton, 2017), a state-of-the-art topic model that implements black-box variational inference (Ranganath et al., 2014), to include contextualized representations. Our approach leads to consistent and significant improvements in two standard metrics on topic coherence and produces competitive topic diversity results.

Contributions We propose a straightforward and easily implementable method that allows neural topic models to create coherent topics. We show

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 759–766
August 1–6, 2021. ©2021 Association for Computational Linguistics

that the use of contextualized document embeddings in neural topic models produces significantly more coherent topics. Our results suggest that topic models benefit from latent contextual information, which is missing in BoW representations. The resulting model addresses one of the most central issues in topic modeling. We release our implementation as a Python library, available at the following link: https://github.com/MilaNLProc/contextualized-topic-models.

2 Neural Topic Models with Language Model Pre-training

We introduce a Combined Topic Model (CombinedTM) to investigate the incorporation of contextualized representations in topic models. Our model is built around two main components: (i) the neural topic model ProdLDA (Srivastava and Sutton, 2017) and (ii) the SBERT embedded representations (Reimers and Gurevych, 2019). Let us notice that our method is indeed agnostic about the choice of the topic model and the pre-trained representations, as long as the topic model extends an autoencoder and the pre-trained representations embed the documents.

ProdLDA is a neural topic modeling approach based on the Variational AutoEncoder (VAE). The neural variational framework trains a neural inference network to directly map the BoW document representation into a continuous latent representation. Then, a decoder network reconstructs the BoW by generating its words from the latent document representation[1]. The framework explicitly approximates the Dirichlet prior using Gaussian distributions, instead of using a Gaussian prior like Neural Variational Document Models (Miao et al., 2016). Moreover, ProdLDA replaces the multinomial distribution over individual words in LDA with a product of experts (Hinton, 2002).

We extend this model with contextualized document embeddings from SBERT (Reimers and Gurevych, 2019),[2] a recent extension of BERT that allows the quick generation of sentence embeddings. This approach has one limitation. If a document is longer than SBERT's sentence-length limit, the rest of the document will be lost. The document representations are projected through a hidden layer with the same dimensionality as the vocabulary size, concatenated with the BoW representation. Figure 1 briefly sketches the architecture of our model. The hidden layer size could be tuned, but an extensive evaluation of different architectures is out of the scope of this paper.

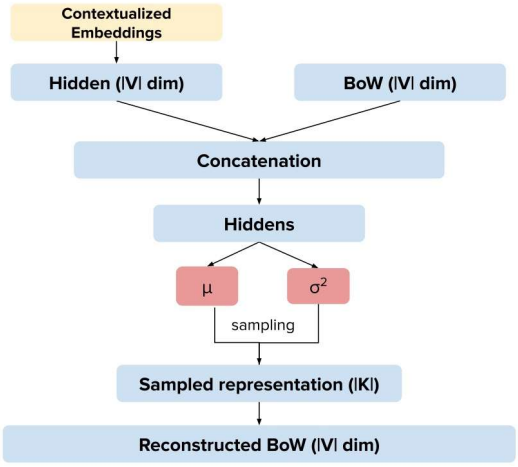


Figure 1: High-level sketch of CombinedTM. Refer to (Srivastava and Sutton, 2017) for more details on the architecture we extend.

Dataset	Docs	Vocabulary
20Newsgroups	18,173	2,000
Wiki20K	20,000	2,000
StackOverflow	16,408	2,303
Tweets2011	2,471	5,098
Google News	11,108	8,110

Table 1: Statistics of the datasets used.

3 Experimental Setting

We provide detailed explanations of the experiments (e.g., runtimes) in the supplementary materials. To reach full replicability, we use open-source implementations of the algorithms.

3.1 Datasets

We evaluate the models on five datasets: 20NewsGroups[3], Wiki20K (a collection of 20,000 English Wikipedia abstracts from Bianchi et al. (2021)), Tweets2011[4], Google News (Qiang et al., 2019), and the StackOverflow dataset (Qiang et al., 2019). The latter three are already pre-processed. We use a similar pipeline for 20NewsGroups and Wiki20K: removing digits, punctuation, stopwords, and infrequent words. We derive SBERT document representations from unpreprocessed text for Wiki20k

[1] For more details see (Srivastava and Sutton, 2017).

[2] https://github.com/UKPLab/sentence-transformers

[3] http://qwone.com/~jason/20Newsgroups/

[4] https://trec.nist.gov/data/tweets/

Model	Avg τ	Avg α	Avg ρ
Results for the Wiki20K Dataset:			
Ours	**0.1823**	0.1980	**0.9950**
PLDA	0.1397	0.1799	0.9901
MLDA	0.1443	**0.2110**	0.9843
NVDM	-0.2938	0.0797	0.9604
ETM	0.0740	0.1948	0.8632
LDA	-0.0481	0.1333	0.9931
Results for the StackOverflow Dataset:			
Ours	**0.0280**	0.1563	0.9805
PLDA	-0.0394	0.1370	**0.9914**
MLDA	0.0136	0.1450	0.9822
NVDM	-0.4836	0.0985	0.8903
ETM	-0.4132	**0.1598**	0.4788
LDA	-0.3207	0.1063	0.8947
Results for the GoogleNews Dataset:			
Ours	**0.1207**	**0.1325**	**0.9965**
PLDA	0.0110	0.1218	0.9902
MLDA	0.0849	0.1219	0.9959
NVDM	-0.3767	0.1067	0.9648
ETM	-0.2770	0.1175	0.4700
LDA	-0.3250	0.0969	0.9774
Results for the Tweets2011 Dataset:			
Ours	**0.1008**	**0.1493**	0.9901
PLDA	0.0612	0.1327	0.9847
MLDA	0.0122	0.1272	**0.9956**
NVDM	-0.5105	0.0797	0.9751
ETM	-0.3613	0.1166	0.4335
LDA	-0.3227	0.1025	0.8169
Results for the 20NewsGroups Dataset:			
Ours	0.1025	0.1715	0.9917
PLDA	0.0632	0.1554	**0.9931**
MLDA	**0.1300**	0.2210	0.9808
NVDM	-0.1720	0.0839	0.9805
ETM	0.0766	**0.2539**	0.8642
LDA	0.0173	0.1627	0.9897

Table 2: Averaged results over 5 numbers of topics. Best results are marked in bold.

and 20NewsGroups. For the others, we use the pre-processed text;[5] See Table 1 for dataset statistics. The sentence encoding model used is the pre-trained RoBERTa model fine-tuned on SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and the STSb (Cer et al., 2017) dataset.[6]

3.2 Metrics

We evaluate each model on three different metrics: two for topic coherence (normalized pointwise mutual information and a word-embedding based measure) and one metric to quantify the diversity of the topic solutions.

Normalized Pointwise Mutual Information (τ) (Lau et al., 2014) measures how related the top-10 words of a topic are to each other, considering the words' empirical frequency in the original corpus. τ is a symbolic metric and relies on co-occurrence. As Ding et al. (2018) pointed out, though, topic coherence computed on the original data is inherently limited. Coherence computed on an external corpus, on the other hand, correlates much more to human judgment, but it may be expensive to estimate.

External word embeddings topic coherence (α) provides an additional measure of how similar the words in a topic are. We follow Ding et al. (2018) and first compute the average pairwise cosine similarity of the word embeddings of the top-10 words in a topic, using Mikolov et al. (2013) embeddings. Then, we compute the overall average of those values for all the topics. We can consider this measure as an external topic coherence, but it is more efficient to compute than Normalized Pointwise Mutual Information on an external corpus.

Inversed Rank-Biased Overlap (ρ) evaluates how diverse the topics generated by a single model are. We define ρ as the reciprocal of the standard RBO (Webber et al., 2010; Terragni et al., 2021b). RBO compares the 10-top words of two topics. It allows disjointedness between the lists of topics (i.e., two topics can have different words in them) and uses weighted ranking. I.e., two lists that share some of the same words, albeit at different rankings, are penalized less than two lists that share the same words at the highest ranks. ρ is 0 for identical topics and 1 for completely different topics.

3.3 Models

Our main objective is to show that contextual information increases coherence. To show this, we compare our approach to ProdLDA (Srivastava and Sutton, 2017, the model we extend)[7], and the

[5] This can be sub-optimal, but many datasets in the literature are already pre-processed.

[6] stsb-roberta-large

[7] We use the implementation of Carrow (2018).

following models: (ii) Neural Variational Document Model (NVDM) (Miao et al., 2016); (iii) the very recent ETM (Dieng et al., 2020), MetaLDA (MLDA) (Zhao et al., 2017) and (iv) LDA (Blei et al., 2003).

3.4 Configurations

To maximize comparability, we train all models with similar hyper-parameter configurations. The inference network for both our method and ProdLDA consists of one hidden layer and 100-dimension of softplus units, which converts the input into embeddings. This final representation is again passed through a hidden layer before the variational inference process. We follow (Srivastava and Sutton, 2017) for the choice of the parameters. The priors over the topic and document distributions are learnable parameters. For LDA, the Dirichlet priors are estimated via Expectation-Maximization. See the Supplementary Materials for additional details on the configurations.

4 Results

We divide our results into two parts: we first describe the results for our quantitative evaluation, and we then explore the effect on the performance when we use two different contextualized representations.

4.1 Quantitative Evaluation

We compute all the metrics for 25, 50, 75, 100, and 150 topics. We average results for each metric over 30 runs of each model (see Table 2).

As a general remark, our model provides the most coherent topics across all corpora and topic settings, even maintaining a competitive diversity of the topics. This result suggests that the incorporation of contextualized representations can improve a topic model's performance.

LDA and NVDM obtain low coherence. This result has also also been confirmed by Srivastava and Sutton (2017). ETM shows good external coherence (α), especially in 20NewsGroups and StackOverflow. However, it fails at obtaining a good τ coherence for short texts. Moreover, ρ shows that the topics are very similar to one another. A manual inspection of the topics confirmed this problem. MetaLDA is the most competitive of the models we used for comparison. This may be due to the incorporation of pre-trained word embeddings into MetaLDA. Our model provides very competitive results, and the second strongest model appears to be MetaLDA. For this reason, we provide a detailed comparison of τ in Table 3, where we show the average coherence for each number of topics; we show that on 4 datasets over 5 our model provides the best results, but still keeping a very competitive score on 20NG, where MetaLDA is best.

Wiki20K	25	50	75	100	150
Ours	**0.17**$^{\clubsuit}$	**0.19**$^{\clubsuit}$	**0.18**$^{\clubsuit}$	**0.19**$^{\clubsuit}$	**0.17**$^{\clubsuit}$
MLDA	0.15	0.15	0.14	0.14	0.13
SO					
Ours	**0.05**	**0.03**$^{\clubsuit}$	**0.02**$^{\clubsuit}$	**0.02**$^{\clubsuit}$	**0.02**$^{\clubsuit}$
MLDA	**0.05**$^{\clubsuit}$	0.02	0.00	-0.02	0.00
GNEWS					
Ours	**-0.03**$^{\clubsuit}$	**0.10**$^{\clubsuit}$	**0.15**$^{\clubsuit}$	**0.18**$^{\clubsuit}$	**0.19**$^{\clubsuit}$
MLDA	-0.06	0.07	0.13	0.16	0.14
Tweets					
Ours	**0.05**$^{\clubsuit}$	**0.10**$^{\clubsuit}$	**0.11**$^{\clubsuit}$	**0.12**$^{\clubsuit}$	**0.12**$^{\clubsuit}$
MLDA	0.00	0.05	0.06	0.04	-0.07
20NG					
Ours	0.12	0.11	0.10	0.09	0.09
MLDA	**0.13**$^{\clubsuit}$	**0.13**$^{\clubsuit}$	**0.13**$^{\clubsuit}$	**0.13**$^{\clubsuit}$	**0.12**$^{\clubsuit}$

Table 3: Comparison of τ between CombinedTM (ours) and MetaLDA over various choices of topics. Each result averaged over 30 runs. $^{\clubsuit}$ indicates statistical significance of the results (t-test, p-value < 0.05).

Readers can see examples of the top words for each model in the Supplementary Materials. These descriptors illustrate the increased coherence of topics obtained with SBERT embeddings.

4.2 Using Different Contextualized Representations

Contextualized representations can be generated from different models and some representations might be better than others. Indeed, one question left to answer is the impact of the specific contextualized model on the topic modeling task. To answer to this question we rerun all the experiments with CombinedTM but we used different contextualized sentence embedding methods as input to the model.

We compare the performance of CombinedTM using two different models for embedding the contextualized representations found in the SBERT repository:[8] *stsb-roberta-large* (Ours-R), as employed in the previous experimental setting, and using *bert-base-nli-means* (Ours-B). The latter is derived from a BERT model fine-tuned on NLI

[8] https://github.com/UKPLab/sentence-transformers

	Wiki20K	SO	GN	Tweets	20NG
Ours-R	**0.18**	**0.03**	**0.12**	**0.10**	**0.10**
Ours-B	**0.18**	0.02	0.08	0.06	0.07

Table 4: τ performance of CombinedTM using different contextualized encoders.

data. Table 4 shows the coherence of the two approaches on all the datasets (we averaged all results). In these experiments, RoBERTa fine-tuned on the STSb dataset has a strong impact on the increase of the coherence. This result suggests that including novel and better contextualized embeddings can further improve a topic model's performance.

5 Related Work

In recent years, neural topic models have gained increasing success and interest (Zhao et al., 2021; Terragni et al., 2021a), due to their flexibility and scalability. Several topic models use neural networks (Larochelle and Lauly, 2012; Salakhutdinov and Hinton, 2009; Gupta et al., 2020) or neural variational inference (Miao et al., 2016; Mnih and Gregor, 2014; Srivastava and Sutton, 2017; Miao et al., 2017; Ding et al., 2018). Miao et al. (2016) propose NVDM, an unsupervised generative model based on VAEs, assuming a Gaussian distribution over topics. Building upon NVDM, Dieng et al. (2020) represent words and topics in the same embedding space. Srivastava and Sutton (2017) propose a neural variational framework that explicitly approximates the Dirichlet prior using a Gaussian distribution. Our approach builds on this work but includes a crucial component, i.e., the representations from a pre-trained transformer that can benefit from both general language knowledge and corpus-dependent information. Similarly, Bianchi et al. (2021) replace the BOW document representation with pre-trained contextualized representations to tackle a problem of cross-lingual zero-shot topic modeling. This approach was extended by Mueller and Dredze (2021) that also considered fine-tuning the representations. A very recent approach (Hoyle et al., 2020) which follows a similar direction uses knowledge distillation (Hinton et al., 2015) to combine neural topic models and pre-trained transformers.

6 Conclusions

We propose a straightforward and simple method to incorporate contextualized embeddings into topic models. The proposed model significantly improves the quality of the discovered topics. Our results show that context information is a significant element to consider also for topic modeling.

Ethical Statement

In this research work, we used datasets from the recent literature, and we do not use or infer any sensible information. The risk of possible abuse of our models is low.

Acknowledgments

We thank our colleague Debora Nozza and Wray Buntine for providing insightful comments on a previous version of this paper. Federico Bianchi and Dirk Hovy are members of the Bocconi Institute for Data Science and Analytics (BIDSA) and the Data and Marketing Insights (DMI) unit.

References

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021. Cross-lingual contextualized topic models with zero-shot learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Stephen Carrow. 2018. PyTorchAVITM: Open Source AVITM Implementation in PyTorch. Github.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 288–296. Curran Associates, Inc.

Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malú Castellanos, and Riddhiman Ghosh. 2013. Discovering coherent topics using general knowledge. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 209–218. ACM.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 795–804. The Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Ran Ding, Ramesh Nallapati, and Bing Xiang. 2018. Coherence-aware neural topic modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 830–836.

Pankaj Gupta, Yatin Chaudhary, Florian Buettner, and Hinrich Schütze. 2019. Document informed neural autoregressive topic models with distributional prior. In *AAAI2019*, pages 6505–6512. AAAI Press.

Pankaj Gupta, Yatin Chaudhary, Thomas Runkler, and Hinrich Schuetze. 2020. Neural topic modeling with continual lifelong learning. In *International Conference on Machine Learning*, pages 3907–3917. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Alexander Miserlis Hoyle, Pranav Goel, and Philip Resnik. 2020. Improving Neural Topic Models using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1752–1771, Online. Association for Computational Linguistics.

Hugo Larochelle and Stanislas Lauly. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012*, pages 2717–2725.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 530–539.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2410–2419. PMLR.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1727–1736. JMLR.org.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Andriy Mnih and Karol Gregor. 2014. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1791–1799. JMLR.org.

Aaron Mueller and Mark Dredze. 2021. Fine-tuning encoders for improved monolingual and zero-shot polylingual neural topic modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3054–3068, Online. Association for Computational Linguistics.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making Sense of Language-Specific BERT Models. *arXiv preprint arXiv:2003.02912*.

Jipeng Qiang, Qian Zhenyu, Yun Li, Yunhao Yuan, and Xindong Wu. 2019. Short text topic modeling techniques, applications, and performance: A survey. *arXiv preprint arXiv:1904.07695*.

Rajesh Ranganath, Sean Gerrish, and David M. Blei. 2014. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 814–822. JMLR.org.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015*, pages 399–408. ACM.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Ruslan Salakhutdinov and Geoffrey E. Hinton. 2009. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 1607–1614. Curran Associates, Inc.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. 2021a. OCTIS: Comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270, Online. Association for Computational Linguistics.

Silvia Terragni, Elisabetta Fersini, and Enza Messina. 2020a. Constrained relational topic models. *Information Sciences*, 512:581–594.

Silvia Terragni, Elisabetta Fersini, and Enza Messina. 2021b. Word embedding-based topic similarity measures. In *Natural Language Processing and Information Systems - 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021*. Springer.

Silvia Terragni, Debora Nozza, Elisabetta Fersini, and Messina Enza. 2020b. Which matters most? comparing the impact of concept and document relationships in topic models. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 32–40.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 4207–4213. ijcai.org.

Yi Yang, Doug Downey, and Jordan L. Boyd-Graber. 2015. Efficient methods for incorporating knowledge into topic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 308–317. The Association for Computational Linguistics.

He Zhao, Lan Du, Wray Buntine, and Gang Liu. 2017. Metalda: A topic model that efficiently incorporates meta information. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 635–644. IEEE.

He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. 2021. Topic modelling meets deep neural networks: A survey. *arXiv preprint arXiv:2103.00498*.

A Datasets

We pre-processed 20NewsGroup and Wiki20K. We removed punctuation, digits, and nltk's English stop-words. Following other researchers, we selected 2,000 as the maximum number of words for the BoW, and thus we kept only the 2,000 most frequent words in the documents. The other datasets come already pre-processed (reference links are in the paper) and thus we take them as is.

B Models and Baselines

B.1 ProdLDA

We use the implementation made available by Carrow (2018) since it is the most recent and with the most updated packages (e.g., one of the latest versions of PyTorch). We run 100 epochs of the model. We use ADAM optimizer. The inference network is composed of a single hidden layer and 100-dimension of softplus units. The priors over

the topic and document distributions are learnable parameters. Momentum is set to 0.99, the learning rate is set to 0.002, and we apply 20% of drop-out to the hidden document representation. The batch size is equal to 200. More details related to the architecture can be found in the original work (Srivastava and Sutton, 2017).

B.2 Combined TM

The model and the hyper-parameters are the same used for ProdLDA with the difference that we also use SBERT features in combination with the BoW: we take the SBERT English embeddings, apply a (learnable) function/dense layer $R^{1024} \rightarrow R^{|V|}$ and concatenate the representation to the BoW. We run 100 epochs of the model. We use ADAM optimizer.

B.3 LDA

We use Gensim's[9] implementation of this model. The hyper-parameters α and β, controlling the document-topic and word-topic distribution respectively, are estimated from the data during training.

B.4 ETM

We use the implementation available at https://github.com/adjidieng/ETM with default hyperparameters.

B.5 Meta-LDA

We use the authors' implementation available at https://github.com/ethanhezhao/MetaLDA. As suggested, we use the Glove embeddings to initialize the models. We used the 50-dimensional embeddings from https://nlp.stanford.edu/projects/glove/. The parameters α and β have been set to 0.1 and 0.01 respectively.

B.6 Neural Variational Document Model (NVDM)

We use the implementation available at https://github.com/ysmiao/nvdm with default hyperparameters, but using two alternating epochs for encoder and decoder.

C Computing Infrastructure

We ran the experiments on two common laptops, equipped with a GeForce GTX 1050: models can be easily run with basic infrastructure (having a GPU is better than just using CPU, but the experiment can also be replicated with CPU). Both laptops have 16GB of RAM. CUDA version for the experiments was 10.0.

C.1 Runtime

What influences the computational time the most is the number of words in the vocabulary. Table 5 shows the runtime for one epoch of both our Combined TM (CTM) and ProdLDA (PDLDA) for 25 and 50 topics on Google News and 20Newsgroups datasets with the GeForce GTX 1050. ProdLDA is faster than our Combined TM. This is due to the added representation. However, we believe that these numbers are quite similar and make our model easy to use, even with common hardware.

	GNEWS		20NG	
	50 topics	100 topics	50 topics	100 topics
CTM	2.1s	2.2s	1.2s	1.2s
PLDA	1.5s	1.5s	0.8s	0.9s

Table 5: Time to complete one epoch on Google News and 20Newsgroups datasets with 25 and 50 topics.

[9] https://radimrehurek.com/gensim/models/ldamodel.html

Input Representations for Parsing Discourse Representation Structures: Comparing English with Chinese

Chunliu Wang
CLCG
Univ. of Groningen
chunliu.wang@rug.nl

Rik van Noord
CLCG
Univ. of Groningen
r.i.k.van.noord@rug.nl

Arianna Bisazza
CLCG
Univ. of Groningen
a.bisazza@rug.nl

Johan Bos
CLCG
Univ. of Groningen
johan.bos@rug.nl

Abstract

Neural semantic parsers have obtained acceptable results in the context of parsing DRSs (Discourse Representation Structures). In particular models with character sequences as input showed remarkable performance for English. But how does this approach perform on languages with a different writing system, like Chinese, a language with a large vocabulary of characters? Does rule-based tokenisation of the input help, and which granularity is preferred: characters, or words? The results are promising. Even with DRSs based on English, good results for Chinese are obtained. Tokenisation offers a small advantage for English, but not for Chinese. Overall, characters are preferred as input, both for English and Chinese.

1 Introduction

Recently, sequence-to-sequence models have achieved remarkable performance in various natural language processing tasks, including semantic parsing (Dong and Lapata, 2016; Jia and Liang, 2016; Konstas et al., 2017; Dong and Lapata, 2018), the task of mapping natural language to formal meaning representations (Figure 1). In this short paper we focus on parsing Discourse Representation Structures (DRSs): the meaning representations proposed in Discourse Representation Theory (DRT, Kamp and Reyle, 1993), covering a large variety of linguistic phenomena including coreference, thematic roles, presuppositions, scope, quantification, tense, and discourse relations.

Several data-driven methods based on neural networks have been proposed for DRS parsing (van Noord et al., 2018b, 2019; Liu et al., 2019a; Evang, 2019; Fancellu et al., 2019; Fu et al., 2020; van Noord et al., 2020). These approaches frame semantic parsing as a sequence transformation problem and map the target meaning representation to string format. These models learn the meaning of a series of semantic phenomena by taking sentences

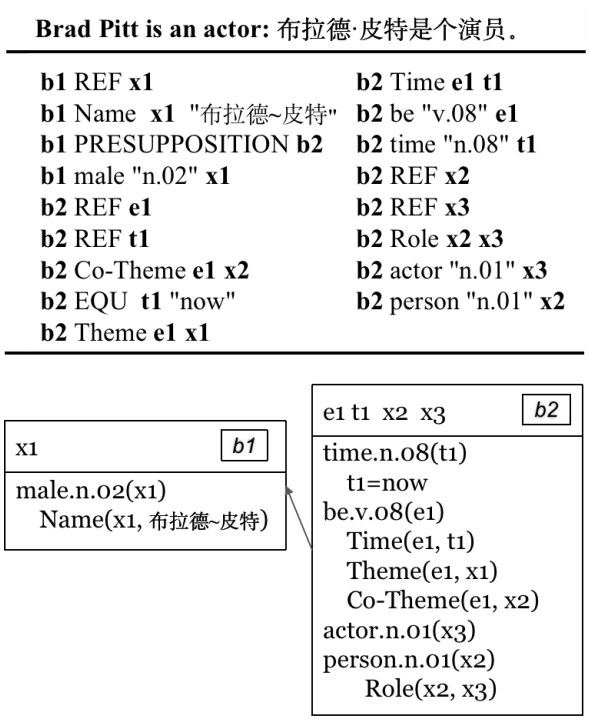


Figure 1: Example DRS for Chinese in both clause and box representation.

as input and directly outputting the corresponding DRSs, without the aid of any extra linguistic information (such as part-of-speech or syntactic structure). These previous studies have achieved good results, but have mostly focused on English or other languages that use the Latin alphabet.

Our objective is to investigate whether the same method is applicable to Mandarin Chinese, an extremely analytic language which makes deep parsing challenging (Levy and Manning, 2003; Yu et al., 2011; Tse and Curran, 2012; Min et al., 2019). But Chinese is not only different on the level of syntax; its writing system also shows large differences with English, as there are no explicit word separators in written Chinese, and there is no distinction between

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 767–775
August 1–6, 2021. ©2021 Association for Computational Linguistics

lower- and upper case characters. Unlike English, Chinese words comprise few characters, but the number of different characters is about two orders of magnitude higher than that of English.

These orthographic differences are interesting in the context of previous work, as van Noord et al. (2018b) use character-level input and word-level input to compare the impact of different input representations on DRSs parsing for English, finding that the character-level representation obtained better performance. In this paper we want to investigate how Chinese fits in this picture. To the best of our knowledge, we are the first to explore methods for Chinese DRS parsing. We aim to answer the following questions:

1. Can existing DRS parsing models achieve good results for Chinese? (RQ1)
2. Given the different writing systems used for English and Chinese, which input granularity is best for either language? (RQ2)
3. Is rule-based word segmentation (tokenization) beneficial for Chinese DRS parsing? (RQ3)

This paper is organised as follows. First we provide a short background on the formal meaning representations that we use, the difference between the writing systems of English and Chinese, and the issues that arise around characters and words. Then we will introduce our approach, the data set that we use, and how we conduct our experiments. In the final section we show that we can achieve good results for Chinese DRS parsing, with characters as the preferred representation.

2 Background

Representing Meaning DRT proposes DRSs to represent the meaning of sentences and short texts. An impressive repertoire of semantic phenomena is covered by DRT, including quantification, negation, reference resolution, comparatives, discourse relations, and presupposition. We use the DRS version as employed in the Parallel Meaning Bank (Abzianidze et al., 2017), where concepts (triggered by nouns, verbs, adjectives and adverbs) are represented by WordNet synsets (Fellbaum, 1998), and semantic relations by Verbnet roles (Kipper et al., 2008). DRSs can be represented in box format or clause format (see Figure 1), where x, e, s, and t are discourse referents denoting individuals, events, states, and time, respectively, and b is used for variables denoting DRSs. Named entities are preserved from the original language used in the input, so names in Chinese are literally transferred in the DRS interpretation (see Figure 1). This means that the only difference between English and Chinese DRSs is the way names are represented: English orthography is used for proper names in English DRSs; Chinese characters are used for names in the corresponding Chinese DRSs.

The box format has become a common representation of DRSs because of its convenient reading and intuitive understanding. The clause format is a flat version of the standard box notation, which represents DRSs as a set of clauses. Due to its simple and flat structure it is more suitable for machine learning purposes. At the same time, however, the structure of DRSs poses a challenge to sequence-to-sequence models, because they need to be able to generate the well-formed recursive semantic structures.

Chinese Word Segmentation Differently from English, Chinese words are not separated by white spaces, as shown in Table 1. The first step of a typical Chinese NLP task is usually to use separators to mark boundaries at appropriate positions to identify words in a sentence. These words define the basic semantic units of Chinese. This process, i.e., Chinese word segmentation (Lafferty et al., 2001; Xue, 2003; Zheng et al., 2013; Cai and Zhao, 2016; Min et al., 2019), is a fundamental step for many Chinese NLP applications, which directly affects downstream performance (Foo and Li, 2004; Xu et al., 2004). Despite the large body of existing research, the quality of Chinese word segmentation remains far from perfect, because many characters are highly ambiguous.

Input Formats for Neural Methods Character-level representations have proved useful for neural network models in many NLP tasks such as POS-tagging (Santos and Zadrozny, 2014; Plank et al., 2016), dependency parsing (Ballesteros et al., 2015) and neural machine translation (Chung et al., 2016). However, only a few studies have used character-level representations as input representations for Chinese NLP tasks (Yu et al., 2017; Li et al., 2018, 2019; Min et al., 2019). For Chinese semantic parsing, previous studies mostly used word-based representations as well (Che et al., 2016; Wang et al., 2018). For English DRS parsing, how-

Type	English input representation	Chinese input representation
Char (raw)	^ b r a d | ^ p i t t | i s | a n | a c t o r .	布 拉 德 · 皮 特 是 个 演 员 。
Char (continuous)	^ b r a d ^ p i t t i s a n a c t o r .	
Char (tokenized)	^ b r a d | ^ p i t t | i s | a n | a c t o r | .	布 拉 德 | · | 皮 特 | 是 | 个 | 演 员 | 。
Word	brad pitt is an actor .	布拉德 · 皮特 是 个 演员 。
BPE (5k)	^ b@ ra@ d ^ p@ it@ t is an ac@ tor@ .	布 拉 德 · 皮 特 是个 演 员。

Table 1: Input representations for the English sentence *Brad Pitt is an actor* and its Chinese translation (布拉德.皮特是个演员). Note that raw and continuous character representations are identical in Chinese. Char (tokenized) adds explicit word boundaries after tokenizing the text. The symbol | represents a word boundary, while the symbol ^ represents a shift to uppercase.

ever, van Noord et al. (2018b) showed that a bi-LSTM sequence-to-sequence model with character-level representations outperformed word-based representations, as well as a combination of words and characters. This will be the starting point of our exploration of Chinese DRS parsing.

3 Methodology

Annotated Data We use data from the Parallel Meaning Bank (PMB 3.0.0, Abzianidze et al., 2017). The documents in this PMB release are sourced from seven different corpora from a wide range of genres. For one of these corpora, Tatoeba, Chinese translations already exist, and we added them to the PMB data. For the remaining texts that had no Chinese translation, we translated the English documents into Chinese using the Baidu API, manually verified the results and, when needed, corrected the translations. Only a few translations needed major corrections. About a hundred translated sentences lacked past or future tense or used uncommon Chinese expressions. Special care was given to the translation of named entities, ambiguous words, and proverbs, and required about a thousand changes. For economical reasons the silver part of the data was only checked on grammatical fluency. Table 2 shows the difference in word- and character-level vocabulary size between English and Chinese. The full translated data set is publicly available.[1]

Language	Chars	Tokens	Words	Tokens
English	139	5,149,912	32,447	1,088,252
Chinese	3,832	1,514,181	39,705	950,310

Table 2: Vocabulary sizes and number of tokens. The number of tokens is calculated after tokenizing the text with either Moses or Jieba.

[1] https://github.com/wangchunliu/Chinese-DRS-data

Chinese Meaning Representations We start from the English–Chinese sentence pairs with the DRSs originally annotated for English. Interestingly, the DRSs in the PMB can be conceived as language-neutral. Even though the English WordNet synsets present in the DRS are reminiscent of English, they really represent concepts, not words. Similarly, the VerbNet roles have English names, but are universal thematic roles. An exception is formed by named entities, that are grounded by the orthography used in the source language. In sum, we assume that the translations are, by and large, meaning preserving, and project English to Chinese DRSs by changing all English named entities to Chinese ones as they appeared in the Chinese input (see Figure 1). This semantic annotation projection method bears strong similarities and is inspired by Damonte et al. (2017) and Min et al. (2019).

Input Representation Types We consider five types of input representations, outlined in Table 1: (i) raw characters, (ii) continuous characters (i.e., without spaces), (iii) tokenised characters, (iv) tokenised words, and (v) byte-pair encoded text (BPE, Sennrich et al., 2016). Note that for Chinese, the first two options amount to the same kind of input. For BPE, we experiment with the number of merges (1k, 5k and 10k) and found in preliminary experiments that it was preferable to not add the indicator "@" for Chinese. For English character input we use an explicit "shift" symbol (^) to indicate uppercased characters, to keep the vocabulary size low. Moreover, the | symbol represents an explicit word boundary. For tokenisation we use the Moses tokenizer (Koehn et al., 2007) for English, while we use the default mode of the Jieba tokenizer[2] to segment the Chinese sentences. To fairly compare these different input representations, we do not employ pretrained embeddings.

[2] https://github.com/fxsjy/jieba

Output Representation Appendix B shows how DRSs are represented for the purpose of training neural models, following van Noord et al. (2018b). Variables are replaced by indices, and the DRSs are coded in either a linearised character-level or word-level clause format. For Chinese, we experimented with both representations and found that the output representation had little effect on parsing performance. To follow previous work (van Noord et al., 2018b) and to allow a fair comparison between the languages, we therefore use the character-level DRS representation for both languages.

Data Splits We distinguish between *gold* (manually corrected meaning representations) and *silver* (automatically generated and partially corrected meaning representations) data. There are a total of 8,403 English–Chinese documents with gold data, of which 885 are used as development set and 898 as test set. The silver data (97,597 documents) is only used to augment the training data, following van Noord et al. (2018b). We use a fine-tuning approach to effectively use high-quality data in our experiments: first training the system with silver and gold data, then restarting the training to fine-tune on only the gold data.

Neural Architecture We use a recurrent sequence-to-sequence neural network with two bi-directional LSTM layers (Hochreiter and Schmidhuber, 1997) as implemented by Marian (Junczys-Dowmunt et al., 2018), similar to van Noord et al. (2019).[3] Specific hyper-parameters are shown in Appendix A. We also experimented with the Transformer model (Vaswani et al., 2017), as implemented in the same framework. However, similar to van Noord et al. (2020), none of our experiments reached the performance of the bi-LSTM model. We will therefore only show results of the bi-LSTM model in this paper.

Evaluation DRS output is evaluated by using Counter (van Noord et al., 2018a), a tool that calculates the micro precision and recall of matching DRS clauses. Counter has been widely used in the evaluation of DRS parsers (Abzianidze et al., 2019). The generated DRSs have to be syntactically as well as semantically well-formed, as checked by the Referee tool (van Noord et al., 2018b), and are otherwise penalised with an F-score of 0.[4]

[3]Code to reproduce our experiments is available at: https://github.com/wangchunliu/Chinese-DRS-parsing

[4]For all our models, this only happened <1% of the time.

	English		Chinese	
Input type	**Dev**	**Test**	**Dev**	**Test**
Char (raw)	87.9	87.6	78.8	76.2
Char (continuous)	86.1	86.9		
Char (tokenised)	88.0	88.1	79.5	76.2
BPE (1k)	86.8	87.0	78.5	76.2
BPE (5k)	87.4	87.1	75.1	71.8
BPE (10k)	82.5	82.3	68.5	65.2
Word	84.5	83.2	74.7	71.6

Table 3: F-scores for DRS parsing with different input representations, averaged over 5 training runs. For BPE, the number of merges is given.

4 Results and Discussion

Table 3 shows the average of five runs for each input representation type. Generally, performance on English is significantly better than on Chinese, which is not surprising as the DRSs are based on English input using English WordNet synsets as concepts (see Figure 1). Given the situation, it is remarkable that Chinese reaches high scores given the differences between the languages in how they convey meaning (Levy and Manning, 2003).

In general, F-scores start to decrease when sentences get longer (Figure 2), though there is no clear difference between the character and word-level models. This is in line with the findings of van Noord et al. (2018b). For English, the input types based on characters outperform those based on words. BPE approaches character-level performance for small amounts of merges (1k), but never surpasses it. This too is in line with van Noord et al. (2018b), but also with previous work on NMT for Chinese (Li et al., 2019). There is a small benefit (0.5) for tokenizing the input text before converting the input to character-level format, though the continuous character representation also works surprisingly well. For Chinese, character-based input shows the best performance too, though for a very small amount of merges BPE obtains a similar score. As opposed to English, tokenizing the Chinese input is not beneficial when using a character-level representation, though it also does not hurt performance. In general, character-level models seem the most promising for Chinese DRS parsing. Similar results were obtained by Min et al. (2019) for Chinese SQL parsing.

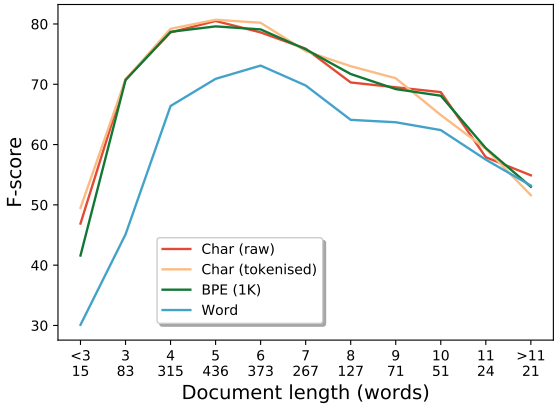


Figure 2: F-scores over Chinese document length on the combined dev and test set, averaged over 5 runs. The x-axis shows the document length in words (top) and the number of documents for that length (bottom).

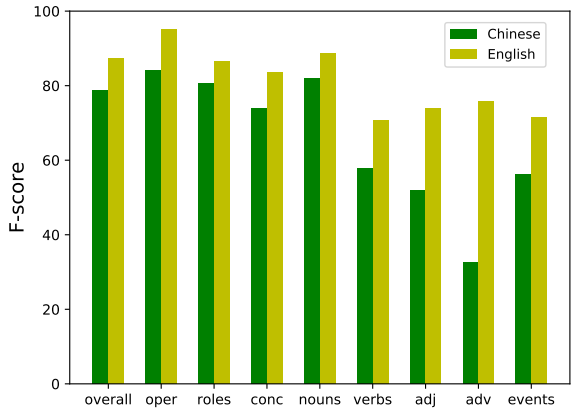


Figure 3: F-scores per clause type (DRS operators, VerbNet roles and WordNet concepts) and concept type (nouns, verbs, adjectives, adverbs and events) as introduced by van Noord et al. (2018b). Reported scores are on the Chinese and English dev set for the raw character-level models, averaged over 5 runs.

Figure 3 shows detailed scores for the character-based (raw) model on the Chinese and English dev set, categorizing operators (e.g., negation, presupposition or modalities), VerbNet roles (e.g., Agent, Theme), predicates, and senses. Modifiers, especially adverbs, get a systematic lower score in Chinese compared to English. This is interesting, and examination of the data reveals that English adverbs are regularly translated as Chinese noun phrases (e.g., *slightly* → *a little*). This will lower the F-score even though the meaning is preserved, only expressed in a semantically different way.

5 Conclusion and Future Work

DRS parsing for Chinese based on projecting meaning representations from English translations gives remarkable performance (**RQ1**), though Chinese adverbs remain challenging. English results outperform those of Chinese, but it is likely that this is due to the general bias of the meaning representations towards English. Similar as for English, we find that characters are the preferred input representation for Chinese (**RQ2**). Surprisingly, for English, good results are even obtained by using characters without spaces as input. Tokenisation (segmenting the text into words) of the input offers a small advantage for English, but not for Chinese (**RQ3**), though it will be interesting to experiment with higher quality word segmentation systems (Higashiyama et al., 2019; Tian et al., 2020).

There are many research directions one could take next. One is to include pre-trained models. For instance, we could use recently proposed pre-trained models such as BART (Lewis et al., 2020) or mBART (Liu et al., 2020) to improve parsing performance. Another interesting idea is, rather than assuming the English WordNet as a background ontology for concepts in the DRS, using concepts based on Chinese WordNet or multilingual wordnets (Wang and Bond, 2013; Bond and Foster, 2013). Both possibilities will likely further improve performance of semantic parsing for Chinese and inspire research for developing semantic parsing models for languages other than English.

Acknowledgements

This work was funded by the NWO-VICI grant "Lost in Translation—Found in Meaning" (288-89-003). The first author is supported by the China Scholarship Council (CSC201904890008). Arianna Bisazza was partly funded by the Netherlands Organization for Scientific Research (NWO) under project number 639.021.646. The Tesla K40 GPU used in this work was kindly donated to us by the NVIDIA Corporation. We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster. Finally, we thank the anonymous reviewers for their insightful comments.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019. The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.

Wanxiang Che, Yanqiu Shao, Ting Liu, and Yu Ding. 2016. SemEval-2016 task 9: Chinese semantic dependency parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1074–1080, San Diego, California. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Federico Fancellu, Sorcha Gilroy, Adam Lopez, and Mirella Lapata. 2019. Semantic graph parsing with recurrent neural network DAG grammars. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2769–2778, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum. 1998. Wordnet: An electronic lexical database. *The MIT Press, Cambridge, Ma., USA*.

Schubert Foo and Hui Li. 2004. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management: an International Journal*, 40(1):p.161–190.

Qiankun Fu, Yue Zhang, Jiangming Liu, and Meishan Zhang. 2020. DRTS parsing with structure-aware encoding and decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6818–6828, Online. Association for Computational Linguistics.

Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto, and Isaac Okada. 2019. Incorporating word attention into character-based word segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2699–2709, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Hans Kamp and U. Reyle. 1993. From discourse to logic: Introduction to model theoretic semantics of natural language, formal logic and discourse representation theory. *Language*, 71(4).

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42:21–40.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289.

Roger Levy and Christopher D. Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 439–446, Sapporo, Japan. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Haonan Li, Zhisong Zhang, Yuqi Ju, and Hai Zhao. 2018. Neural character-level dependency parsing for Chinese. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5205–5212. AAAI Press.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of Chinese representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252, Florence, Italy. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019a. Discourse representation parsing for sentences and documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6248–6262, Florence, Italy. Association for Computational Linguistics.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019b. Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for Chinese SQL semantic parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3652–3658, Hong Kong, China. Association for Computational Linguistics.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018a. Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018b. Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord, Antonio Toral, and Johan Bos. 2019. Linguistic information in neural semantic parsing with multiple encoders. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. Character-level representations improve DRS-based semantic parsing Even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4587–4603, Online. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.

Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st international conference on machine learning (ICML-14)*, pages 1818–1826.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020. Improving Chinese word segmentation with wordhood memory networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online. Association for Computational Linguistics.

Daniel Tse and James R. Curran. 2012. The challenges of parsing Chinese with Combinatory Categorial Grammar. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 295–304, Montréal, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Chuan Wang, Bin Li, and Nianwen Xue. 2018. Transition-based Chinese AMR parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 247–252, New Orleans, Louisiana. Association for Computational Linguistics.

Shan Wang and Francis Bond. 2013. Building the Chinese open Wordnet (COW): Starting from core synsets. In *Proceedings of the 11th Workshop on Asian Language Resources*, pages 10–18, Nagoya, Japan. Asian Federation of Natural Language Processing.

Jia Xu, Richard Zens, and Hermann Ney. 2004. Do we need Chinese word segmentation for statistical machine translation? In *Proceedings of the Third SIGHAN Workshop on Chinese Language Processing*, pages 122–128, Barcelona, Spain. Association for Computational Linguistics.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, pages 29–48.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint embeddings of Chinese words, characters, and fine-grained subcharacter components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 286–291, Copenhagen, Denmark. Association for Computational Linguistics.

Kun Yu, Yusuke Miyao, Takuya Matsuzaki, Xiangli Wang, and Junichi Tsujii. 2011. Analysis of the difficulties in Chinese deep parsing. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 48–57, Dublin, Ireland. Association for Computational Linguistics.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.

A Hyperparameters

Table 4 gives an overview of the hyperparameters we experimented with in the tuning stage. The hyperparameters of the bi-LSTM model are mostly taken from van Noord et al. (2018b), but tuned on the Chinese development set. The hyperparameters of the Transformer model were randomly selected, and then also tuned on the Chinese development set. We also experimented with the hyperparameter selection of Liu et al. (2019b) for the Transformer model, but did not get the desired results.

Fine-tuning We first train the models on gold + silver data for 15 epochs, then we restart the training process from that checkpoint to fine-tune on only the gold data for 30 epochs.

bi-LSTM					
Parameter	**value**	**Parameter**	**value**	**Parameter**	**value**
dim-emb	300	dim-rnn	300	enc-cell	lstm
dec-cell	lstm	enc-depth	2	dec-depth	2
mini-batch	32	lr-decay	0.5	lr-decay-strategy	epoch
normalize	0.9	beam-size	10	learn-rate	0.002
dropout-rnn	0.2	cost-type	ce-mean	label-smoothing	0.1
optim	adam	early-stop	3	valid-metric	cross-entropy
Transformer					
enc-depth	2	dec-depth	2	transformer-aan-depth	2
lr-decay	0.8	optim	adam	transformer-ffn-depth	2
dropout	0.1	dim-emb	300	transformer-dim-ffn	256
num-heads	4	normalize	0.6	transformer-dim-aan	256
label-smoothing	0.1	beam-size	10	learn-rate	0.0002
mini-batch	32	lr-decay-strategy	epoch	valid-metric	cross-entropy

Table 4: Hyperparameters setting for the Marian bi-LSTM model and Transformer model.

B Output representation

Figure 4 shows two possible DRS representations for the output of our models, which were introduced by van Noord et al. (2018b). Here, we show an example of a Chinese input sentence and corresponding Chinese DRS representations. In this paper, we use the character-level representation.

Input sentence

汤姆提着一桶水。

Output DRS

```
b1 REF x1                  b2 Time e1 t1              b2 REF x2
b1 Name x1 "汤姆"          b2 time "n.08" t1          b2 bucket "n.01" x2
b1 PRESUPPOSITION b2       b2 REF e1                  b2 Content x2 x3
b1 male "n.02" x1          b2 Agent e1 x1             b2 REF x3
b2 REF t1                  b2 Theme e1 x2             b2 water "n.06" x3
b2 TPR t1 "now"            b2 carry "v.01" e1
```

Char-level DRS

```
$NEW ||| REF *** $0 ||| Name ||| @0 ||| " 汤 姆 " *** $0 ||| PRESUPPOSITION ||| $NEW *** $-1 ||| m a l e ||| " n . 0 2 " ||| @0 *** $0 ||| REF *** $0 ||| TPR ||| @0 ||| "now" *** $0 ||| Time ||| @1 ||| @0 *** $0 ||| t i m e ||| " n . 0 8 " ||| @0 *** $0 ||| REF *** $0 ||| Agent ||| @0 ||| @-2 *** $0 ||| Theme ||| @0 ||| @1 *** $0 ||| c a r r y ||| " v . 0 1 " ||| @0 *** $0 ||| REF *** $0 ||| b u c k e t ||| " n . 0 1 " ||| @0 *** $0 ||| Content ||| @0 ||| @1 *** $0 ||| REF *** $0 ||| w a t e r ||| " n . 0 6 " ||| @0
```

Word-level DRS

```
$NEW REF *** $0 Name @0 "汤姆" *** $0 PRESUPPOSITION $NEW *** $-1 male "n.02" @0 *** $0 REF *** $0 TPR @0 "now" *** $0 Time @1 @0 *** $0 time "n.08" @0 *** $0 REF *** $0 Agent @0 @-2 *** $0 Theme @0 @1 *** $0 carry "v.01" @0 *** $0 REF *** $0 bucket "n.01" @0 *** $0 Content @0 @1 *** $0 REF *** $0 water "n.06" @0
```

Figure 4: The result of preprocessing a DRS to a character-level and word-level representation, respectively.

Code Generation from Natural Language with Less Prior and More Monolingual Data

Sajad Norouzi* **Keyi Tang** **Yanshuai Cao**
Borealis AI
sajadn@cs.toronto.edu,{keyi.tang,yanshuai.cao}@borealisai.com

Abstract

Training datasets for semantic parsing are typically small due to the higher expertise required for annotation than most other NLP tasks. As a result, models for this application usually need additional prior knowledge to be built into the architecture or algorithm. The increased dependency on human experts hinders automation and raises the development and maintenance costs in practice. This work investigates whether a generic transformer-based seq2seq model can achieve competitive performance with minimal code-generation-specific inductive bias design. By exploiting a relatively sizeable monolingual corpus of the target programming language, which is cheap to mine from the web, we achieved 81.03% exact match accuracy on Django and 32.57 BLEU score on CoNaLa. Both are SOTA to the best of our knowledge. This positive evidence highlights a potentially easier path toward building accurate semantic parsers in practice. †

1 Introduction

For a machine to act upon users' natural language inputs, a model needs to convert the natural language utterances to machine-understandable meaning representation, i.e. semantic parsing (SP). The output meaning representation is beyond shallow identification of topic, intention, entity or relation, but complex structured objects expressed as logical forms, query language or general-purpose programs. Therefore, annotating parallel corpus for semantic parsing requires more costly expertise.

SP shares some resemblance with machine translation (MT). However, SP datasets are typically smaller, with only a few thousand to at most tens of thousands of examples, even smaller than most low resource MT problems. Simultaneously, because the predicted outputs generally need to be exactly correct to execute and produce the right answer, the accuracy requirement is generally higher than MT. As a result, inductive bias design in architecture and algorithm has been prevalent in the SP literature (Dong and Lapata, 2016; Yin and Neubig, 2017, 2018; Dong and Lapata, 2018; Guo et al., 2019; Wang et al., 2019; Yin and Neubig, 2019).

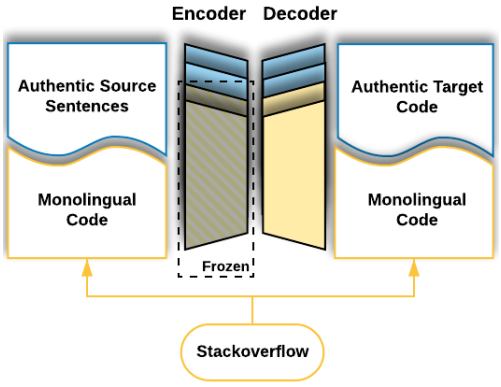


Figure 1: TAE: the monolingual corpus is used both as source and target. The encoder is frozen in the computation branch on the monolingual data.

While their progress is remarkable, excessive task-specific expert design makes the models complicated, hard to transfer to new domains, and challenging to deploy in real-world applications. In this work, we look at the opposite end of the spectrum and try to answer the following question: *with little inductive bias in the model, and no additional labelled data, is it still possible to achieve competitive performance?* This is an important question, as the answer could point to a much shorter road to practical SP without breaking the bank.

This paper shows that the answer is encouragingly affirmative. By exploiting a relatively large monolingual corpus of the programming language, a transformer-based Seq2Seq model (Vaswani et al., 2017) with little SP specific prior could potentially attain results superior to or competitive with the

*Work done during internship at BorealisAI
†Code at https://github.com/BorealisAI/code-gen-TAE

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 776–785
August 1–6, 2021. ©2021 Association for Computational Linguistics

state-of-the-art models specially designed for semantic parsing. Our contributions are three-fold:

- We provide evidence that transformer-based seq2seq models can reach a competitive or superior performance with models specifically designed for semantic parsing. This suggests an alternative route for future progress other than inductive bias design;
- We do empirical analysis over previously proposed approaches for incorporating monolingual data and show the effectiveness of our modified technique on a range of datasets;
- We set the new state-of-the-art on Django (Oda et al., 2015) reaching 81.03% exact match accuracy and on CoNaLa (Yin et al., 2018) with a BLEU score of 32.57.

2 Previous Work on Semantic Parsing

Different sources of prior knowledge about the SP problem structure could be exploited.

Input structure: Wang et al. (2019) adapts the transformer relative position encoding (Shaw et al., 2018) to express relations among the database schema elements as well as with the input text spans. Herzig and Berant (2020) proposed a span-based neural parser with compositional inductive bias built-in. Herzig and Berant (2020) also leverages a CKY-style (Cocke, 1969; Kasami, 1966; Younger, 1967) inference to link input features to output codes.

Output structure: The implicit tree or graph-like structures in the programs can also be exploited. Dong and Lapata (2016) proposed parent-feeding LSTM following the tree structure. Dong and Lapata (2018) proposed a coarse-to-fine decoding approach. Guo et al. (2019) crafted an intermediate meaning representation to bridge the large gap between input utterance and the output SQL queries. Yin and Neubig (2017, 2018) proposed TranX, a more general-purpose transition-based system, to ensure grammaticality of predictions. Using TranX, the neural model predicts the linear sequence of AST-tree constructing actions instead of the program tokens. However, a human expert needs to craft the grammar, and the design quality impacts the learning and generalization for the neural nets.

Sequential models with less SP specific priors have been investigated (Dong and Lapata, 2016; Ling et al., 2016b; Zeng et al., 2020), However, they generally fell short in accuracy comparing to the best of structure-exploiting ones listed above.

The most closely related to ours is the work by Xu et al. (2020) for incorporating external knowledge from extra datasets, which used a noisy parallel dataset from Stackoverflow to pre-train the SP and fine-tuned it on the primary dataset. Their approach's main limitation is still the need for (noisy) parallel data, albeit cheaper than the primary labelled set. Nonetheless, as we shall see in the experiment section later, our approach achieves better results when using the same amount of data mined from the same source despite ignoring the source sentence.

3 Background and Methodology

BERT (Devlin et al., 2018) class of pre-trained models can make up for the lack of inductive bias on the input side to some degree. On the output side, we hope to learn the necessary prior knowledge about the target meaning representation from unlabelled monolingual data.

Using monolingual data to improve seq2seq models is not new and has been extensively studied in MT before. Notable methods include *fusion* (Gulcehre et al., 2015; Ramachandran et al., 2016; Sriram et al., 2018; Stahlberg et al., 2018), *back-translation (BT)* (Sennrich et al., 2015; Edunov et al., 2018; Hoang et al., 2018), (Currey et al., 2017; Burlot and Yvon, 2018, 2019), and *BT with copied monolingual data* (Currey et al., 2017; Burlot and Yvon, 2019). However, due to more structured outputs, less training data, and different evaluation metrics of exact match correctness instead of BLEU, it is unclear if these lessons transfer from MT to SP. So SP-specific investigation is needed.

3.1 Target Autoencoding with Frozen Encoder

We assume a parallel corpus of natural language utterances and their corresponding programs, $\mathcal{B} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$. The goal is to train a translator model (TM) to maximize the conditional log probability of $\boldsymbol{y}_i$ given $\boldsymbol{x}_i$, $T_{\boldsymbol{\theta}}(\boldsymbol{y}_i|\boldsymbol{x}_i)$, over the training set: $\mathcal{L}_{\text{sup}} = \sum_{\mathcal{B}} T_{\boldsymbol{\theta}}(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$ where $\boldsymbol{\theta}$ is the vector of TM model parameters. Let $\mathcal{M} = \{\boldsymbol{y}'_i\}$ denote the monolingual dataset in the target language.

Currey et al. (2017); Burlot and Yvon (2019) demonstrated that in low resource MT, auto-encoding the monolingual data besides the main supervised training is helpful. Following the same path, we add an auto-encoding objective term on monolingual data: $\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{sup}} + \sum_{\mathcal{M}} T_{\boldsymbol{\theta}}(\boldsymbol{y}'_i \mid \boldsymbol{y}'_i)$.

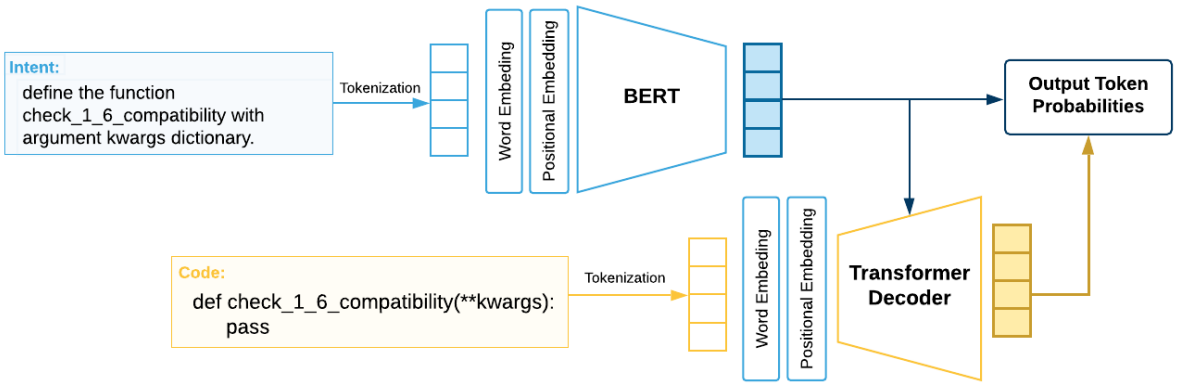


Figure 2: Model overview during training: we use a standard transformer-based encoder-decoder model where the positional and word embeddings are shared between encoder and decoder. The modules related to the encoder are represented in blue and the decoder ones are in yellow. Standard teacher forcing and transformer masking is applied during training.

The target $\boldsymbol{y}_i'$'s are reconstructed using the shared encoder-decoder model.

We conjecture that monolingual data auto-encoding mainly helps the decoder, so we propose to freeze the encoder parameters for monolingual data. Writing the encoder and decoder parameters separately with $\boldsymbol{\theta} = [\boldsymbol{\theta}_e, \boldsymbol{\theta}_d]$, then $\boldsymbol{\theta}_e$ is updated using the gradient of the supervised objective $\mathcal{L}_{\text{sup}}$, whereas the decoder gradient comes from $\mathcal{L}_{\text{full}}$. We verify this hypothesis in section 4.1.

In terms of model architecture, our TM is a standard transformer-based seq2seq model with copy attention (Gu et al., 2016) (illustrated in Fig. 2 of C). We fine-tune BERT as the encoder and use a 4-layer transformer decoder. There is little SP-specific inductive bias in the architecture. The only special structure is the copy attention, which is not a strong inductive bias designed for SP as copy attention is widely used in other tasks as well.

We refer to the method of using copied monolingual data and freezing the encoder over them as *target autoencoding* (TAE). Unless otherwise specified in the ablation studies, the encoder is always frozen.

4 Experiments

For our primary experiments we considered two python datasets namely Django and CoNaLa. The former is based on Django web framework and the latter is annotated code snippets from stackoverflow answers. Additionally, we experiment on the SQL version of GeoQuery and ATIS from Finegan-Dollak et al. (2018) (with query split), WikiSQL (Zhong et al., 2017), and Magic (Java) (Ling et al., 2016b).

Python Monolingual Corpora: CoNaLa comes with $600K$ mined questions from Stackoverflow. We ignored the noisy source intents/sentences and just use the python snippets. To be comparable with Xu et al. (2020), we also select a corresponding $100K$ subset version for comparison. See Appendix A for details on the SQL and Java monolingual corpora.

Experimental Setup: In all experiments, we use label smoothing with a parameter of 0.1 and Polyak averaging (Polyak and Juditsky, 1992) of parameters with a momentum of 0.999 except for GeoQuery which we use 0.995. We use Adam (Kingma and Ba, 2014) and early stopping based on the dataset specific evaluation metric on dev set. The learning rate for the encoder is 1×10^{-5} over all datasets. We used the learning rate of 7.5×10^{-5} on all datasets except GeoQuery and ATIS which we use $1 \times 10 - 4$. The architecture overview is shows in Fig. 2. At the inference time we use beam search with beam size of 10 and a length normalization based on (Wu et al., 2016). We run each experiment with 5 different random seeds and report the average and standard deviation. WordPiece tokenization is used for both natural language utterances and programming code.

4.1 Empirical Analysis

First, we considered a scenario where the monolingual corpus comes from the same distribution as the bitext. We simulate this setup by using 10% of Django training data as labeled data while using all the python examples from Django as the mono-

Source:	call the function lazy with 2 arguments : _string_concat and six.text_type , substitute the result for string_concat .	Source:	define the function timesince with d , now defaulting to none, reversed defaulting to false as arguments .
Gold & TAE:	`string_concat = lazy(_string_concat, six.text_type)`	Gold & TAE:	`def timesince(d, now=none, reversed=false):` `  pass`
Baseline:	`string_concat = lazy (_concat_concat , six.text_type )`	Baseline:	`def timesince (d = none, reversed ( d = false ):` `  pass`
Note:	copy mistake: wrong variable resulting from failed copy	Note:	unbalanced paranthesis and multiple semantic mistakes.

Table 1: Example mistakes by the baseline that are fixed by TAE. More examples in Appendix E.

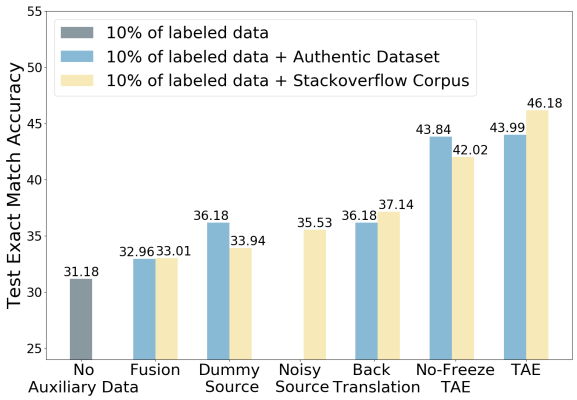


Figure 3: Analysis using only 10% Django train bitext.

lingual dataset of 10 times bigger. Results with "Authentic Dataset" in Fig. 3 shows the effectiveness of TAE vs other approaches.

Next, we used the monolingual dataset prepared for python (StackOverflow Corpus) which is from a different distribution. Fig. 3 shows even more considerable improvement, thanks to the larger monolingual set. We considered noisy intents provided in CoNaLa monolingual corpus and dummy source sentences where each monolingual sample is paired along with a random length array containing zeros. We also compared against other well-known approaches like fusion and back-translation, see experiments details in Appendix D. TAE outperforms all those approaches by a large margin.

Now one important question is, what part of the model benefits from monolingual data most? In Sec. 3.1, we conjectured that auto-encoding of monolingual data should mostly help the decoder, not the encoder. To verify this, we perform an ablation by comparing freezing encoder parameters versus not freezing over the monolingual set. Fig. 3 shows that without freezing the encoder, performance drops slightly for TAE on authentic Django while dropping significantly when copying on Stackoverflow data. This confirms that the performance gain is due to its effect on the decoder, while the copied monolingual data might even hurts the encoder.

4.2 Main Results on Full Data

Table 2-3 showcase our SOTA results on Django and CoNaLa. While our simple base seq2seq model does not outperform previous works, with TAE on the monolingual data, our performance improves and outperforms all the previous works.

The most direct comparison is with Xu et al. (2020) that also leverage the same extra data mined from StackOverflow (*EK* in Table 3). As mentioned in Sec. 2, they used the noisy parallel corpus for pre-training, whereas we only leverage the monolingual set. However, we obtain both larger relative improvements over our baseline (32.29 from 30.98) compared to Xu et al. (2020) (28.14 from 27.20), as well as better absolute results in the best case. In fact, with only the $100K$ StackOverflow monolingual data, our result is on par with the best one from Xu et al. (2020) that uses the additional python API bitext data. Finally, note that part of our superior performance is due to using BERT as an encoder.

Finally, TAE also yields improvements on other programming languages, as shown for GeoQuery (SQL), ATIS (SQL) and Magic (Java) in Table 4. We observe no improvement on WikiSQL. But it is not surprising given its large dataset size and the simplicity of its targets. As observed by previous works (Finegan-Dollak et al., 2018), more than half of queries follow simple pattern of "`SELECT col FROM table WHERE col = value`".

The main results in terms of improvement over previous best methods are statistically significant in Table 2-3. On Django, our result is better than Reranker (Yin and Neubig, 2019) (best previous method in Table 2) with a P-value < 0.05, under one-tailed two-sample t-test for mean equality. Since the previous state of the art on CoNaLa (EK + 100k + API in Table 3) did not provide the standard deviation, we cannot conduct a two-sample t-test against it. Instead, we performed a one-tailed two-sample t-test against the TranX+BERT baseline and observed that our improvement is statistically significant with P-value < 0.05. In Table 4,

Model	Django
YN17 (Yin and Neubig, 2017)	71.6
TRANX (Yin and Neubig, 2018)	73.7
Coarse2Fine (Dong and Lapata, 2018)	74.1
TRANX2 (Yin and Neubig, 2019)	77.3 ± 0.4
TRANX2 + BERT	79.7 ± 0.42
Reranker (Yin and Neubig, 2019)*	80.2 ± 0.4
Our baseline	77.05 ± 0.6
Our baseline + TAE	$\mathbf{81.03 \pm 0.14}$

Table 2: Exact match accuracy for Django test set. Yin and Neubig (2019)* trained a separate model on top of SP to rank beam search outputs.

Model	CoNaLa
Reranker (Yin and Neubig, 2019)*	30.11
TRANX (Yin and Neubig, 2019) + BERT	30.47 ± 0.7
EK (baseline) (Xu et al., 2020)	27.20
EK + 100k (Xu et al., 2020)	28.14
EK + 100k + API (Xu et al., 2020)*†	32.26
Our baseline	30.98 ± 0.1
Our baseline + TAE on 100k	32.29 ± 0.4
Our baseline + TAE on 600k	$\mathbf{32.57 \pm 0.3}$

Table 3: CoNaLa test BLEU. Methods with * trained a separate model on top of SP to rerank beam search outputs. Xu et al. (2020)† used an additional bitext corpus mined from python API documentation.

Dataset	Baseline (%)	Baseline + TAE (%)
GeoQuery	47.69 ± 0.05	51.87 ± 0.02
ATIS	38.04 ± 0.77	40.56 ± 0.57
Magic	41.61 ± 2.07	42.34 ± 0.52
WikiSQL	85.36 ± 0.06	85.30 ± 0.07

Table 4: Additional dataset results: test set exact match accuracy on all dataset.

improvements on GeoQuery and ATIS are statistically significant with P-value < 0.05, while it is not the case for Magic and WikiSQL.

4.3 Discussion

Thus far, we have verified that the decoder benefits from TAE and the encoder does not. For a better understanding of what TAE improves in the decoder, we propose two metrics namely *copy accuracy* and *generation accuracy*. Copy accuracy only considers tokens appearing in the source sentence. If the model produces all of the tokens that need to be copied from the source sentence, and in the right order, then the score is one otherwise zero for the example. Generation-accuracy ignores tokens appearing in the source intent and computes the exact match accuracy of the prediction. We show how to compute these metrics for the following example:
Question: define the function timesince with d, now defaulting to none, reversed defaulting to false as arguments.
Ground Truth:
"`def timesince(d, now=none, reversed=false): pass`"

We iterate over the ground truth script tokens one by one and remove those that can be copied from the source, leading to this code:
Generation Ground Truth:
"`def (=none=):pass`", and the removed tokens will be considered for copy ground truth.
Copy Ground Truth: "`timesince d , now , reversed false`".

We would then use the copy and generation ground truth strings to compute each metric. Note that the order of tokens are still important and exact equality is required.

As shown in Table 5 both metrics are improved. Table 1 illustrates one example from each type and with more samples in the Appendix E. Copy accuracy is important for producing the right variable names mentioned, and it is improved as expected. It is also encouraging to see quantitatively and qualitatively that grammar mistakes are reduced, meaning that the lack of prior knowledge of target language structure is compensated by learning from monolingual data.

Model	Copy	Generation
10% basline	34.18	55.73
10% baseline + TAE	58.89	66.31
Full baseline	80.11	81.27
Full baseline + TAE	84.59	82.65

Table 5: Copy and generation accuracies on Django test set

5 Conclusion

This work has shown the possibility to achieve a competitive or even SOTA performance on semantic parsing with little or no inductive bias design. Besides the usual large-scale pre-trained encoders, the key is to exploit relatively large monolingual corpora of the meaning representation. The modified copied monolingual data approach from machine translation literature works well in this extremely low-resource setting. Our results point to a promising alternative direction for future progress.

Acknowledgements

We appreciate the ACL anonymous reviewers and area chair for their valuable inputs. We also would like to thank a number of Borealis AI colleagues for helpful discussions, including Wei (Victor) Yang, Peng Xu, Dhruv Kumar, and Simon J. D. Prince for feedback on the writing.

References

Miltiadis Allamanis and Charles Sutton. 2013. Mining Source Code Repositories at Massive Scale using Language Modeling. In *The 10th Working Conference on Mining Software Repositories*, pages 207–216. IEEE.

Franck Burlot and François Yvon. 2019. Using monolingual data in neural machine translation: a systematic study. *arXiv preprint arXiv:1903.11437*.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. *ArXiv*, abs/1903.11437.

John Cocke. 1969. *Programming languages and their compilers: Preliminary notes*. New York University.

Anna Currey, Antonio Valerio Miceli-Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Catherine Finegan-Dollak, Jonathan K Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-sql evaluation methodology. *arXiv preprint arXiv:1806.09029*.

Jiatao Gu, Z. Lu, Hang Li, and V. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *ArXiv*, abs/1603.06393.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-sql in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535.

J. Herzig and J. Berant. 2020. Span-based semantic parsing for compositional generalization. *arXiv preprint arXiv:2009.06040*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Tadao Kasami. 1966. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016a. Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 599–609, Berlin, Germany. Association for Computational Linguistics.

Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, Andrew Senior, Fumin Wang, and Phil Blunsom. 2016b. Latent predictor networks for code generation. *arXiv preprint arXiv:1603.06744*.

Yusuke Oda, Hiroyuki Fudaba, Graham Neubig, Hideaki Hata, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Learning to generate pseudo-code from source code using statistical machine translation. In *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, ASE '15, pages 574–584, Lincoln, Nebraska, USA. IEEE Computer Society.

Boris T Polyak and Anatoli B Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855.

Prajit Ramachandran, Peter J Liu, and Quoc V Le. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Anuroop Sriram, Heewoo Jun, S. Satheesh, and A. Coates. 2018. Cold fusion: Training seq2seq models together with language models. *ArXiv*, abs/1708.06426.

Felix Stahlberg, J. Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. *ArXiv*, abs/1809.00125.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2019. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*.

Y. Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, G. Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. S. Corrado, Macduff Hughes, and J. Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Frank F Xu, Zhengbao Jiang, Pengcheng Yin, Bogdan Vasilescu, and Graham Neubig. 2020. Incorporating external knowledge through pre-training for natural language to code generation. *arXiv preprint arXiv:2004.09015*.

Ziyu Yao, Daniel S Weld, Wei-Peng Chen, and Huan Sun. 2018. Staqc: A systematically mined question-code dataset from stack overflow. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1693–1703. International World Wide Web Conferences Steering Committee.

Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. 2018. Learning to mine aligned code and natural language pairs from stack overflow. In *International Conference on Mining Software Repositories*, MSR, pages 476–486. ACM.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. *arXiv preprint arXiv:1704.01696*.

Pengcheng Yin and Graham Neubig. 2018. Tranx: A transition-based neural abstract syntax parser for semantic parsing and code generation. *arXiv preprint arXiv:1810.02720*.

Pengcheng Yin and Graham Neubig. 2019. Reranking for neural semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4553–4559.

Daniel H Younger. 1967. Recognition and parsing of context-free languages in time n3. *Information and control*, 10(2):189–208.

Jichuan Zeng, Xi Victoria Lin, S. Hoi, R. Socher, Caiming Xiong, Michael R. Lyu, and Irwin King. 2020. Photon: A robust cross-domain text-to-sql system. *ArXiv*, abs/2007.15280.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

A Datasets

We used 6 datasets in total. Django includes programs from Django web framework and CoNaLa contains diverse set of intents annotated on python snippets gathered from Stackoverflow. WikiSQL, GeoQuery, and ATIS include natural language questions and their corresponding SQL queries. WikiSQL includes single table queries while GeogQuery and ATIS requires queries on more than one table. Finally, Magic has Java class implementation of game cards with different methods used during the game. Table 6 summarises all the parallel datasets. For GoeQuery we used query split provided by (Finegan-Dollak et al., 2018).

Monolingual Corpus: CoNaLa comes with $600K$ mined questions from Stackoverflow. We ignored the noisy source intents/sentences and just use the python snippets. To be comparable with Xu et al. (2020), we also select a corresponding $100K$ subset version for comparison. For SQL, Yao et al. (2018) automatically parsed StackOverflow questions related to SQL and provided a set containing $120K$ SQL examples. We automatically parsed the SQL codes and removed samples with grammatical mistakes. We also filtered samples not starting with SELECT special token. Allamanis and Sutton (2013) downloaded full repositories of individual projects that were forked at least once; duplicate projects were removed. We randomly sampled 100K Java examples from more than 14K projects and use that as monolingual set. Table 7 summarises all the monolingual datasets.

Parallel Corpus	Language	Train	Dev	Test
Django (Oda et al., 2015) (link)	Python	16000	1000	1805
CoNaLa (Yin et al., 2018) (link)	Python	2,179	200	500
WikiSQL (Zhong et al., 2017) (link)	SQL	56,355	8421	15878
ATIS (Finegan-Dollak et al., 2018) (link)	SQL	4812	121	347
GeoQuery (Finegan-Dollak et al., 2018) (link)	SQL	536	159	182
Magic (Ling et al., 2016a) (link)	Java	8,457	446	483

Table 6: Parallel dataset sizes. We filtered out Magic data with java code longer than 350 tokens in order to fit in GPU memory.

Monolingual Corpus	Source	Size
Python (Yin et al., 2018) (link)	Stackoverflow	100K
SQL (Yao et al., 2018) (link)	Stackoverflow	52K
Java (Allamanis and Sutton, 2013) (link)	Github	100k

Table 7: Monolingual dataset sizes.

B Dev Set Results

Dataset	Baseline (%)	Baseline + TAE (%)
CoNaLa	32.43 ± 0.21	34.81 ± 0.36
ATIS	5.79 ± 0.29	7.23 ± 0.45
GeoQuery	53.33 ± 1.47	52.58 ± 0.70
Django	75.52 ± 0.21	78.56 ± 0.39
Magic	42.26 ± 1.42	44.17 ± 0.99
WikiSQL	85.92 ± 0.09	85.83 ± 0.07

Table 8: Dev set exact match accuracy on all datasets except CoNaLa which uses BLEU. We followed (Yin and Neubig, 2018) implementation of BLEU score which can be found here.

C Architecture and Experiment Details

We selected the decoder learning rate based on linear search over $[1 \times 10^{-3} - 2.5 \times 10^{-5}]$. Number of decoder layers has been decided based on search over {2, 3, 4, 5, 6} layers and 4 layer decoder shows superior performance (we used a single run for hyperparameter selection). Each model has 150M parameters optimized using a single GTX 1080 Ti GPU. With batch size of 16 each step takes 1.7s on GeoQuery dataset (other datasets have very similar runtime). On Django and CoNaLa, we followed (Yin and Neubig, 2018; Xu et al., 2020) on replacing quoted values with a "str#" where # is a unique id. On Magic dataset, we replaced all newline "\n" tokens with "#"; following (Ling et al., 2016a), we splitted Camel-Case words (e.g., class TirionFordring → class Tirion Fordring) and all punctuation characters. We filtered out Magic data with java code longer than 350 tokens in order to fit in GPU memory.

D Back-Translation and Fusion details

For fusion we follow equation 1 where TM stands for translation model and LM stands for language model. τ limits the confidence of the language model and λ controls the balance between TM and LM. figure 4 shows the performance of a base TM trained on 10% of Django training data with test exact match accuracy of 31.80 over different values of λ and τ. The LM is trained over full Django training set.

$$\log p(y_i^t) = \log p_{TM}(y_i^t) + \lambda \log p_{LM}(y_i^t) = \log p_{TM}(y_i^t) + \lambda \log \frac{e^{l_i^t}/\tau}{\sum_i e^{l_i^t}/\tau} \tag{1}$$

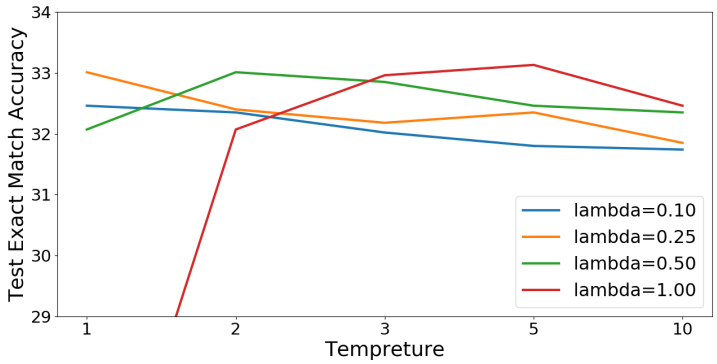


Figure 4: Test exact match accuracy of TM leverage fusion with different parameters

For back-translation we first trained the model using the same architecture explained above in the backward direction. We used BLEU score as a evaluation metric and use early stopping based on that. Using greedy search we generate the corresponding source intent for each code snippet. In the end, the synthetic data is merged with the bitext and trained a forward model.

E Additional Qualitative Examples

Source:	call the function lazy with 2 arguments : _string_concat and six.text_type [six . text_type] , substitute the result for string_concat .
Gold:	`string_concat = lazy(_string_concat, six.text_type )`
Baseline:	`string_concat = lazy (_concat_concat , six. text_type )`
TAE:	`string_concat = lazy ( _string_concat , six. text_type )`
Note:	wrong var

Source:	get translation_function attribute of the object t , call the result with an argument eol_message , substitute the result for result .
Gold:	`result = getattr(t, translation_function)( eol_message)`
Baseline:	`result = getattr ( t , translation_message ) ( eol_message )`
TAE:	`result = getattr ( t , translation_function ) ( eol_message )`
Note:	wrong var

Source:	convert whitespace character to unicode and substitute the result for space .
Gold:	`space = unicode(' ')`
Baseline:	`space = unicode ( character )`
TAE:	`space = unicode ( ' ' )`
Note:	wrongly copied variable name

Source:	assign integer 2 to parts if third element of version equals to zero , otherwise assign it integer 3 .
Gold:	`parts = 2 if version[2] == 0 else 3`
Baseline:	`parts [ 2 ] = 2`
TAE:	`parts = 2 if version [ 2 ] == 0 else 3`
Note:	baseline failed to copy a few source tokens, and instead formed a grammatically correct but semantically incorrect output

Copy mistake examples

Source:	define the function timesince with d , now defaulting to none , reversed defaulting to false as arguments .
Gold:	`def timesince(d, now=none, reversed=false): pass`
Baseline:	`def timesince ( d = none, reversed ( d = false ) : pass`
TAE:	`def timesince ( d, now = none, reversed = false ) : pass`
Note:	unbalanced paranthesis and multiple semantic mistakes.

Source:	define the function exec with 3 arguments : _code_ , _globs_ set to none and _locs_ set to none .
Gold:	`def exec_(_code_, _globs_=none, _locs_=none): pass`
Baseline:	`def exec ( _code_ , _globs= none , _locs_ set ( ) ) : pass`
TAE:	`def exec ( _code_ , _globs_ = none , _locs_ = none ) : pass`
Note:	wrong variable name and grammar mistake

Source:	return an instance of escapebytes , created with an argument , reuslt of the call to the function bytes with an argument s .
Gold:	`return escapebytes(bytes(s))`
Baseline:	`return escapebytes ( bytes ( s ) . re ( s )`
TAE:	`return escapebytes ( bytes ( s ) )`
Note:	extra semantically incorrect predictions and unbalanced paratheses

Source:	call the function blankout with 2 arguments : p and str0 , write the result to out .
Gold:	`out.write(blankout(p, 'str0'))`
Baseline:	`out .write ( blankout ( p , 'str0' )`
TAE:	`out .write ( blankout ( p , 'str0' ) )`
Note:	unbalanced paratheses

Grammar or semantic mistake examples

Table 9: Mistake examples

Issues with Entailment-based Zero-shot Text Classification

Tingting Ma[1,2]*, **Jin-Ge Yao**[2], **Chin-Yew Lin**[2], **Tiejun Zhao**[1]
[1]Harbin Institute of Technology, Harbin, China
[2]Microsoft Research Asia
hittingtingma@gmail.com
{jinge.yao, cyl}@microsoft.com, tjzhao@hit.edu.cn

Abstract

The general format of natural language inference (NLI) makes it tempting to be used for zero-shot text classification by casting any target label into a sentence of hypothesis and verifying whether or not it could be entailed by the input, aiming at generic classification applicable on any specified label space. In this opinion piece, we point out a few overlooked issues that are yet to be discussed in this line of work. We observe huge variance across different classification datasets amongst standard BERT-based NLI models and surprisingly find that pre-trained BERT without any fine-tuning can yield competitive performance against BERT fine-tuned for NLI. With the concern that these models heavily rely on spurious lexical patterns for prediction, we also experiment with preliminary approaches for more robust NLI, but the results are in general negative. Our observations reveal implicit but challenging difficulties in entailment-based zero-shot text classification.

1 Introduction

Natural language inference (NLI, Bowman et al., 2015), also known as recognizing *textual entailment* (RTE, Condoravdi et al., 2003; Dagan et al., 2005), is normally formatted as the task of determining whether or not a **premise** sentence semantically entails a **hypothesis** sentence. The generality of the task format has aroused some recent studies to apply NLI models for various downstream applications (Poliak et al., 2018), and more recently text classification (Yin et al., 2019, 2020), making them generally-applicable solutions along with all those similar attempts to build a universal framework for various NLP tasks (Kumar et al., 2016; Raffel et al., 2020, *inter alia*). Text classification is then reduced to textual entailment by setting the input sentence as the premise and simultaneously casting the candidate label into a hypothesis sentence using pre-defined templates or lexical definitions from WordNet. Once we have any pre-trained NLI models at hand, *zero-shot text classification* under any specified label space is enabled for free without the need to collect annotated data. With contextualized representation based on pre-trained language models such as BERT (Devlin et al., 2019), NLI performance has been drastically improved. Promising empirical results have been shown on various text classification benchmarks that vary across topic classification, emotion classification, and situation classification, outperforming earlier standard approaches (Chang et al., 2008) or simple scoring schemes derived from distributional similarity (Mikolov et al., 2013).

However, such generality is conceptually contradictory with the specificity of text classification in many practical scenarios. In this opinion piece, we conduct extended analysis on the recent attempts (Yin et al., 2019) and point out some implicit issues under entailment-based zero-shot text classification that are overlooked in this line of work. We experiment with additional classification datasets and observe huge variance across them amongst standard BERT-based NLI models. More surprisingly, we find that raw BERT models without fine-tuning can sometimes yield more competitive results. We also experiment with preliminary approaches for improving the robustness of NLI models, but only to find negative results in general. Our observations reveal implicit but massive difficulties in building a successful general-purpose zero-shot text classifier based on text entailment models.

2 Our Investigation and Implied Issues

We attempt at re-examining the earlier study (Yin et al., 2019) with extended analysis to help estab-

*Work during internship at Microsoft Research Asia.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 786–796
August 1–6, 2021. ©2021 Association for Computational Linguistics

lish a better understanding of zero-shot text classification based on textual entailment. Our focus is to check **how well the models pre-trained for NLI could generalize to the prediction of unseen categories**, which is the major target of zero-shot classification. We did not study the setting that test set also include labels seen in training, commonly phrased as *generalized zero-shot learning* (Xian et al., 2018) and referred to as the *label-partially-unseen* setting by Yin et al. (2019). That setting strongly assumes that a bunch of in-domain data for a number of classes are available already. [1]

2.1 Basic setup

2.1.1 Text classification datasets

As an attempt to study zero-shot text classification in conceptually different and diverse aspects, Yin et al. (2019) experimented with three instances:

Topic classification : The Yahoo! Answers dataset from Zhang et al. (2015) with 10 categories.

Emotion classification : The Unify Emotion dataset (Bostan and Klinger, 2018) with 9 emotion types and a `none` label if no emotion applies.

Situation classification : The Situation Typing dataset (Mayhew et al., 2019) with 11 situation types and instances and an extra type `none`.

Additionally, we extend our experiments with the test sets from the following datasets:

Snips : A popular dataset[2] for *intent detection* collected from the Snips personal voice assistant (Coucke et al., 2018), with seven intent labels.

AG's news : To further study the models on *topic classification* in a different genre, we additionally use the English news data from (Zhang et al., 2015) that consists of four types of articles: `World`, `Sports`, `Business`, `Sci/Tech`.

SST-2 : The Stanford Sentiment Treebank dataset[3] processed by Socher et al. (2013) for *sentiment polarity classification* with binary labels (`positive` and `negative`).

[1] Another reason for not studying on this setting is that the split of development set and test set in (Yin et al., 2019) contain the same label space, which is flawed to be used for any claim on the performance of "unseen labels".

[2] https://github.com/snipsco/snips-nlu

[3] For SST-2 we follow Zhang et al. (2021) and Gao et al. (2021) to use the development set from GLUE for testing.

2.1.2 Experimented systems

To study entailment-based approaches, we use the models released by Yin et al. (2019) which are `bert-base-uncased` models pretrained on GLUE RTE (Dagan et al., 2005; Wang et al., 2019b), MNLI (Williams et al., 2018), and FEVER (Thorne et al., 2018), respectively. We reuse the same scheme for mapping labels into hypotheses using templates and WordNet definition for all datasets[4], as well as the same mechanism for producing final predictions. We leave more implementation details to the Appendix.

We keep reporting results from these baselines following Yin et al. (2019) for reference:

- **Majority**: Output the most frequent label.
- **Word2Vec**: Using the average word embeddings to vectorize input and labels, output label with maximum cosine similarity.
- **ESA**: Representing the text and label in the Wikipedia concept vector space. Using the implementation[5] from Chang et al. (2008).

Moreover, due to the obvious variance in performance for models trained on different NLI datasets, we are also tempted to check how much the performance might degrade when given no NLI data at all for fine-tuning. This corresponds to naively using a raw BERT model which has been pre-trained for *next sentence prediction* (NSP). For consistency, we use the same premises and hypotheses as the delegate for label names and templates to formulate the sentence pair classification. Since NSP is not predicting for a directional semantic entailment, we also try a variant with all pairs reversed, i.e., setting all hypothesis sentences ahead of premises as input, denoted as NSP(Reverse).

2.2 Results and further analysis

The results from all systems on different datasets are displayed in Table 1, including an additional group for MNLI results as we found an even better run overall in our experiments. There are some interesting observations emerge from our extended experiments and analysis.

[4] For newly introduced datasets we follow the similar strategy to prepare for the hypothesis templates.

[5] https://github.com/CogComp/cogcomp-nlp/tree/master/dataless-classifier

	Topic (Yahoo)	Emotion	Situation	AG's News	SST-2	Snips
Majority	10.0	5.9	11.0	25.0	50.9	17.7
ESA	28.6	8.0	26.0	73.3	55.5	63.4
Word2Vec	35.7	6.9	15.6	44.1	53.7	63.6
RTE (Yin et al., 2019)	43.8	12.6	**37.2**	56.7	52.5	56.4
FEVER (Yin et al., 2019)	40.1	**24.7**	21.0	**78.3**	71.7	69.4
MNLI (Yin et al., 2019)	37.9	22.3	15.4	72.4	67.5	77.6
MNLI (our best overall run)	49.1	19.9	14.5	77.7	67.5	77.6
NSP (Reverse)	**53.1**	16.1	19.9	**78.3**	**79.7**	**81.3**
NSP	50.6	16.5	25.8	72.1	73.9	73.4

Table 1: Text classification results. We report label-wise weighted F1 for emotion and situtation datasets, and accuracy for the others. Reported results from (Yin et al., 2019) have been reproduced from their released models.

2.2.1 How much have NLI data contributed?

The big difference from various NLI datasets drives us to try a raw BERT without fine-tuning on any NLI data, i.e., merely relying on NSP pre-training for sentence pair classification. The results are shown at the bottom two rows in Table 1, which turn out to be surprisingly strong, especially on topic classification, intent classification, and binary sentiment classification.

We conjecture that the raw BERT model has already acquired certain ability of topic distinction and sentiment polarity due to the construction of positive and negative sentence pairs in NSP pre-training to detect pairwise coherence. In this way, NSP could serve as a non-trivial, strong alternative baseline for zero-shot text classification scenarios where the target labels are semantically more concrete (e.g., topics) or more frequently appeared (e.g., words expressing sentiment). In such scenarios, fine-tuning on limited NLI data could weaken the semantic coherence acquired from the raw BERT pre-trained on generic-domain corpora, especially now that fine-tuned models have utilized many spurious lexical cooccurrence features as shown in many similar sentence pair classification models (Feng et al., 2019; Niven and Kao, 2019), possibly due to the inherent lexical bias from the current NLI datasets collected from crowd workers. [6] Readers who are curious about more details on this problem can refer to our qualitative analysis in the Appendix which could hopefully help establish a slightly better sense on the behavioral difference introduced by NLI fine-tuning.

On the other hand, fine-tuning on NLI data might seem to be marginally helpful for more abstract cases such as emotion and situation typing, but the performance metrics are in fact pathetically disappointing across all systems.

2.2.2 How stable are these NLI models?

Apart from the obvious difference caused by different training data, there underlies a more serious concern: the *discrepancy* between the training task (NLI) and the target usage (classification). The gap in task formatting (and henceforth data distribution) naturally raises a question: *do NLI models with similar in-domain performance generalize similarly for text classification?*

We train NLI models on the largest MNLI dataset with varied hyperparameter settings and random seeds, and keep models achieving similarly strong in-domain generalization performance as measured by the early-stopping dev set performance. Results are listed in Table 2, where the absolute differences between the worst and the best are large, especially on classifying topic or intent. We observe even worse trends on other smaller NLI datasets (see Appendix). These results are consistent with recent studies within the scope of NLI reporting that BERT instances which achieve similar performance metrics on standard NLI datasets could have huge variance in out-of-distribution generalization or linguistic stress testing (McCoy et al., 2020; Zhou et al., 2020; Geiger et al., 2020), while providing another instance of the underspecification problem in modern machine learning (D'Amour et al., 2020).

As a verification, we also try to tune the models for different development sets that better characterize the generalization behavior for zero-shot

[6] Some readers might guess that other NLI datasets collected via a more careful process (Jiang and de Marneffe, 2019; Eisenschlos et al., 2021) might partially mitigate the bias appearing from crowdsourced annotation, but this does not mean that such better intended datasets can be free from statistically biased lexical distributions with coincidental cooccurrences that could be utilized by our strong data-fitting models during fine-tuning (Geirhos et al., 2020; Du et al., 2021). Our additional results described in the Appendix do not seem to be promising on this direction towards better NLI data.

Dataset	Average	Std	Min	Max
MNLI dev set	90.5	0.3	90.0	90.8
Yahoo	39.0	10.5	26.9	50.2
Emotion	18.1	2.0	15.7	20.5
Situation	16.2	1.5	14.5	18.7
AGNews	63.7	11.0	50.0	77.7
SST-2	68.6	2.0	66.1	70.9
Snips	74.1	3.9	68.4	77.6

Table 2: Results of five runs of BERT fine-tuned on MNLI and tested on classification datasets

classification. We reorganize the splitted development set and the test set of the topic classification datasets (Yahoo and AG's News) to make sure they do not have overlapped classes.[7] The new results are shown in Table 3, where we can clearly see more stable generalization performance. This observation necessitates that a certain amount of annotated data for targeted classification already existed, making NLI models difficult to apply in practice. Results in this part reveals that text classification via NLI is asking for out-of-distribution generalization, a property that current NLI models rarely have, henceforth susceptible to huge *instability*.

Dataset	Average	Std	Min	Max
Yahoo-dev	52.7	2.6	49.1	56.2
Yahoo-test	48.1	2.7	44.2	51.7
AGNews-dev	79.0	6.9	72.1	89.1
AGNews-test	73.8	3.8	69.6	77.4

Table 3: Results of five runs for training BERT on MNLI with model selection via target domain dev set

2.2.3 Is more robust NLI helpful?

Previous studies have raised concerns on that the current NLI models heavily rely on spurious lexical overlap patterns (Sanchez et al., 2018; Naik et al., 2018; McCoy et al., 2019, *inter alia*). For analytical purposes, we randomly permute the tokens of each input instance to see how much the predictions might change. Results shown in Table 4 suggest that shuffling the input tokens does not affect the model performance by much, which is consistent with similar recent findings (Gupta et al., 2021; Sinha et al., 2021). This reveals a concern that all these models might just predict with shallow lexical patterns that may not be robust for more semantically abstractive input instances.

There have been a few recent attempts trying to remove the shallow overlap bias for NLI model

[7]Details are described in the Appendix.

Model	Yahoo	AGNews	SST-2
NSP(Reverse)	-5.1 / 67.2	+0.4 / 82.7	-13.5 / 75.9
RTE	-2.0 / 77.5	+0.3 / 90.0	+0.6 / 94.5
FEVER	-7.2 / 64.6	+0.5 / 90.6	-9.5 / 82.3
MNLI	+1.6 / 54.8	+2.7 / 84.9	-6.4 / 84.4
Random	- / 10.0	- / 25.0	- / 50.0

Table 4: Results of shuffling perturbation. In each cell: the change of accuracy after input shuffling, followed by the percentage of examples where the predictions do not change. All these results are reported as the average score of five different random shuffles.

training. We experiment with three schemes on the MNLI data to see whether they could lead to better generalization of zero-shot classification: (1) *Data augmentation* with syntactic transformations (Min et al., 2020)[8], denoted as *DA*, (2) *Instance reweighting* following Clark et al. (2019) that reweights each example with one minus the probability a bias-only model assigns the correct label, denoted as *RW*, and (3) The *bias product* method (Clark et al., 2019) that ensembles a bias-only model via a product of experts, denoted as *BP*, which is essentially the same as its concurrent work via fitting the residual of the biased models (He et al., 2019). There exist additional solutions with richer details such as multi-task learning (Tu et al., 2020) where proper auxiliary tasks could be identified to improve robustness. We plan to explore more in this line in our more extensive future study.

The results are shown in Table 5 . All the three debiasing methods improve the NLI performance on the HANS dataset (McCoy et al., 2019) for robustness testing, indicating that the debiased models overcome the word overlap heuristics to some extend. In general, we do not observe any real improvement other than the neglectable gains on emotion and situation datasets where the original performance is pathetically low.

	HANS	Yahoo	Emo.	Situ.	AG	SST	Snips
MNLI	53.0	**49.1**	19.9	14.5	**77.7**	67.5	**77.6**
w/ DA	**67.3**	47.3	18.0	16.3	74.3	**73.1**	76.6
w/ RW	64.5	43.4	21.8	**23.5**	71.7	68.6	71.6
w/ BP	65.4	48.5	**23.0**	22.3	75.6	69.8	72.7

Table 5: Results of NLI debiasing based on MNLI

[8]We directly use the data released at https://github.com/Aatlantise/syntactic-augmentation-nli

3 Conclusion and Discussion

We investigate entailment-based zero-shot text classification further with extended analysis, uncovering the following overlooked issues:

- Raw BERT models trained for next sentence prediction are surprisingly strong baselines and NLI fine-tuning does not bring performance gain on many classification datasets.
- Large variance on different classification scenarios and instability to different runs, still requiring annotated data (at least used for validation) to stablize generalization performance.
- NLI models usually rely heavily on shallow lexical patterns, which hampers generalization as required by text classification, and currently more robust NLI methods might not help.

Our observations reveal *implicit but massive difficulties in building a usable zero-shot text classifier based on text entailment models*. Given the difficulty of NLI data collection that aims at out-of-domain generalization or transfer learning (Bowman et al., 2020), we question the feasibility of this setup in the current progress of language technology. Before significant progress in language understanding and reasoning, it seems more promising to consider alternative schemes built on explicit external knowledge (Zellers and Choi, 2017; Rios and Kavuluru, 2018; Zhang et al., 2019) or more crafted usage of pre-trained models that hopefully have captured more comprehensive semantic coverage and better compositionality from large corpora or grounded texts (Meng et al., 2020; Brown et al., 2020; Radford et al., 2021).

This study also implies *the huge difficulty for benchmarking zero-shot text classification without any further restriction on the task setting*. The three datasets used by Yin et al. (2019) were originally intended for diverse coverage but are not sufficient to draw consistent conclusions as we have shown. We suggest future studies on zero-shot text classification either conduct experiments over even more diverse classification scenarios to verify any claimed generality, or directly focus on more specific task settings and verify claims within a smaller but clearer scope such as zero-shot intent classification or zero-shot situation typing for more reliable results with less instability, and perhaps based on more carefully curated data (Rogers, 2021).

Acknowledgments

We thank all the anonymous reviewers for their helpful comments on our submitted draft. The empirical studies conducted in this work were mostly based on the open-source repositories on GitHub from other papers as described earlier. We thank their original authors for sharing their implementation and we also publicly release our experimental scripts on GitHub[9].

References

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.

Samuel R. Bowman, Jennimaria Palomaki, Livio Baldini Soares, and Emily Pitler. 2020. New protocols and negative results for textual entailment data collection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8203–8214, Online.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS 2020*.

Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI'08*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

[9] https://github.com/mtt1998/issues-nli

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, pages 177–190. Springer.

Alexander D'Amour, Katherine A. Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yi-An Ma, Cory Y. McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2020. Underspecification presents challenges for credibility in modern machine learning. *CoRR*, abs/2011.03395.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Julian Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan Boyd-Graber. 2021. Fool me twice: Entailment from Wikipedia gamification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 352–365, Online. Association for Computational Linguistics.

Shi Feng, Eric Wallace, and Jordan Boyd-Graber. 2019. Misleading failures of partial-input baselines. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5533–5538, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL-IJCNLP 2021*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. BERT & family eat word salad: Experiments with text understanding. In *35th AAAI Conference on Artificial Intelligence (AAAI-21)*.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Nanjiang Jiang and Marie-Catherine de Marneffe. 2019. Evaluating BERT for natural language inference: A case study on the CommitmentBank. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6086–6091, Hong Kong, China.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 1378–1387.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124.

Stephen Mayhew, Tatiana Tsygankova, Francesca Marini, Zihan Wang, Jane Lee, Xiaodong Yu, Xingyu Fu, Weijia Shi, Zian Zhao, Wenpeng Yin, Karthikeyan K, Jamaal Hay, Michael Shur, Jennifer Sheffield, and Dan Roth. 2019. University of Pennsylvania LoReHLT 2019 Submission. Technical report.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017, Online. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krüger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *OpenAI Technical Report*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, Brussels, Belgium. Association for Computational Linguistics.

Anna Rogers. 2021. Changing the world by changing the data. In *ACL-IJCNLP 2021*.

Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of NLI models: Uncovering the influence of three factors on robustness. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1975–1985, New Orleans, Louisiana.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. Unnatural language inference. *arXiv preprint arXiv:2101.00010*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS 2019*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China.

Wenpeng Yin, Nazneen Fatema Rajani, Dragomir Radev, Richard Socher, and Caiming Xiong. 2020. Universal natural language processing with limited annotations: Try few-shot textual entailment as a start. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8229–8239, Online.

Rowan Zellers and Yejin Choi. 2017. Zero-shot activity recognition with verb attribute induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 946–958, Copenhagen, Denmark. Association for Computational Linguistics.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1031–1040, Minneapolis, Minnesota.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS 2015*.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

A Appendix

A.1 Additional Experimental Details

Templates for generating hypothesis For Yahoo, Emotion, and Situation datasets, we followed Yin et al. (2019) and just explored the label names and WordNet definition accompanied with a template[10] to convert labels to hypotheses for entailment-based models. When applying NSP, we only used label names to generate hypotheses as we did not observe real improvement from using WordNet definitions in our preliminary experiments. For AGNews, SST-2, and Snips, we simply used the label names to fill the templates. The templates we used are given in Table A.1.

Other implementation details For all experiments, we train BERT models by using *bert-base-uncased* version and code from the HuggingFace library (Wolf et al., 2019). We used the same prediction strategy as Yin et al. (2019): we pick the label with the maximal probability in single-label scenarios while choosing all the labels with "next sentence" decision in multi-label cases for both NSP and NSP(Reverse) baselines.

Label spaces of classification The labels of each dataset we used are listed in Table A.2.

[10] https://github.com/yinwenpeng/BenchmarkingZeroShot

Dataset	Template	Label to words mapping
Yahoo	It is related with [LABEL] .	[Sports]: sports, [Society & Culture]: society or culture, etc.
Emotion	This person feels [LABEL] .	[sadness]: sad, [anger]: angry, [guilt]: guilty, etc.
Situation	The people there need [LABLE] .	[shelter]: shelter, [utilities]: utilities, etc.
AGNews	It is related with [LABEL] .	[Sci/Tec]: technology, [Business]: business, etc.
SST-2	The movie is [LABEL] .	[positive]: great , [negative]: terrible
Snips	I want to [LABEL] .	[RateBook] : rate a book, [SearchCreativeWork]: search creative work, etc.

Table A.1: Templates used for each dataset. For Topic Emotion and Situation dataset, we also use the WordNet definitions following Yin et al. (2019)

Dataset	Labels
Yahoo	Society & Culture, Science & Mathematics, Health, Education & Reference, Computers & Internet, Sports, Business & Finance, Entertainment & Music, Family & Relationships, Politics & Government
Emotion	sadness, joy, anger, disgust, fear, surprise, shame, guilt, love, none
Situation	search, evacuate, infrastructure, utilities, water, shelter, medical assistance, food, crimeviolence, terrorism, regime change, none
AGNews	World, Sports, Business, Sci/Tech.
SST-2	Positive, Negative
Snips	RateBook, SearchScreeningEvent, PlayMusic, GetWeather, SearchCreativeWork, AddToPlaylist, BookRestaurant,

Table A.2: The label names of the evaluation datasets.

Additional results on CommitmentBank We finetune BERT on the CommitmentBank dataset (de Marneffe et al., 2019; Wang et al., 2019a) converted into the NLI format (Jiang and de Marneffe, 2019), denoted as CB. Following Wang et al. (2019a), we also try to pretrain BERT on MNLI dataset before finetuning on CommitmentBank, called MNLI+CB. In our experiments, we found both two models trained on CB did not show a better performance compared to model trained on other NLI datasets, especially on Yahoo and AGNews (19.9% accuracy on Yahoo for CB and 17.8% accuracy on Yahoo for MNLI+CB). This indicates that the finetuned BERT models may still focus on features that are beneficial for NLI performance, while losing the topic discriminability.

A.2 Qualitative Analysis

Table 1 shows that NSP(reverse) achieves better performance than NSP on several datasets. This could be related to the templates we used for generating previous or next sentences. For example, for the input *"play the god that failed on vimeo"* with label "PlayMusic", NSP(Reverse) predicts "PlayMusic" while NSP predicts "AddToPlaylist". It is a more natural expression for *"I want to play music. play the god that failed on vimeo"* than *"play the god that failed on vimeo. I want to play music"*. Among the entailment models, We find the RTE-based model performs best on situation dataset. The main class of situation dataset is the "none" label. As shown in Figure A.1, we find RTE-based model performs best on "none" label. Actually, if we calculate the average number of prediction labels each instance, we find NSP, NSP(Reverse), and FEVER's average prediction label number per instance is about 6.2 to 8.3, while RTE and MNLI's average number is about 1, which is closer to the average number of gold labels per instance. The implies NSP is not good at identifying the "none" label since the condition of predicting "entailment" (a premise entails its hypothesis) is more strict than predicting a "next sentence" label. For SST-2, we observe that all three entailment models tend to mislabel sentences with "negative" label as "positive". This may be attributed to the label word distribution in NLI datasets. We find the keyword "great" for positive label is much more frequently occurred than the keyword "terrible" for negative label in all the three NLI datasets.

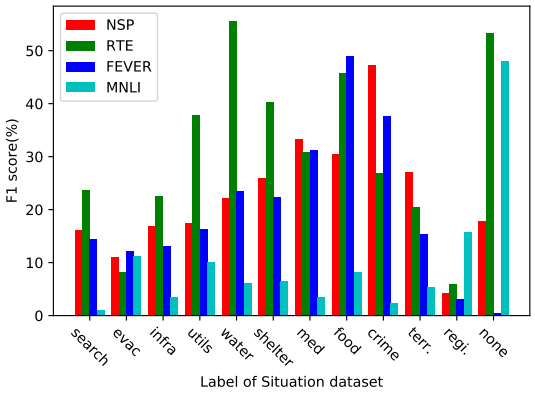


Figure A.1: F1 score of each label in Situation dataset

Case study To get a better understanding of NLI models' behavior, we carry out a case study on

SNIPS. We use Integrated Gradient (Sundararajan et al., 2017) method to attribute the entailment class's output score of BERT model to per input token [11]. Several examples are shown in Table A.3.

We found the NLI models sometimes rely on spurious patterns to do prediction. In the first example, the model finetuned on FEVER assigns a high negative attribution score to the word "*zero*" and makes a wrong prediction. However, if we replace "*zero*" with other numbers, the model changes its prediction and can correctly predicts the "RateBook" label. These examples reflect model trained on FEVER dataset learns the spurious correlations between "not entailment" label and the occurrence of word "*zero*"[12]. These superficial patterns may not be the models' main behaviour for prediction, it still leaks the model's fragility and could be an important factor to the model's failure in zero-shot scenario.

The other two groups of cases show another problem: current NLI models only predict "*entailment*" label when the premise *entails* its hypothesis, this problem definition is just different from the zero-shot test tasks. For example, in the last group, model trained on MNLI outputs a low probability for entailment since "*restaurant*" can not be directly inferred from premise sentence. If we change "*restaurant*" into "*place*", the model confidently predicts "*entailment*" .

Error cases We also show some additional examples in Table A.4, from which we might naturally conjecture that the entailment models could rely on spurious lexical features for prediction.

Impact of template choice How to properly choose templates is another issue when utlizing NLI for zero-shot classification. As shown in Table A.5, different templates that seem meaningful to human might have large performance variance on SST-2.

A.3 Details for Stability Experiments

Details for training settings For MNLI dataset, we merge the *neutral* and *contradiction* labels into *not-entailment* label following Yin et al. (2019). We choose hyperparameters randomly for different runs: we choose learning rate from $\{2e^{-5}, 3e^{-5}, 5e^{-5}\}$, training epochs from $\{3, 4, 5\}$ and randomly set the random seed.

Results for training on RTE As shown in Table A.6, the performance of different runs has large variance on both RTE dev and text classification datasets due to its small size.

Reorganize dev and test sets for Yahoo and AGNews We reorganize the Yahoo development set provided by Yin et al. (2019) and divide test set as follows: For the dev set, the instances with label in set {"Society & Culture", "Health", "Computers & Internet", "Business & Finance", "Family & Relationships"} are preserved, we call this new dev set as **Yahoo-dev** . For the original test set, we only select instances with the label which doesn't appear in the dev set as our new test set, denoted as **Yahoo-test**. During the NLI model training, we select the checkpoint by the performance on Yahoo-dev, and we report the variance of five different runs trained on MNLI. We also conduct experiments on AGNews in the same way. We use {"World", "Sports"} as seen labels and select 1800 instances per seen label randomly in train data as our new development set. In the same way, we get dev set : **AGNews-dev** and our test set **AGNews-test**.

A.4 Details of Robust NLI models

Details for training settings For all the models, we use the same set of hyperparameters: We train all the models with batch size of 64, the Adam optimizer with the initial learning rate of $2e^{-5}$ and finetune the BERT model for 3 epochs. The maximum sequence length is limited to 128.

For DA (data augmentation) method, we use the most effective strategy which is called *inversion with a transformed hypothesis* in Min et al. (2020) . For the bias model used in Reweight and BiasProduct, we use the feature based word overlap bias model[13] in Clark et al. (2019).

Detailed results on HANS Table A.7 shows detailed results for the base BERT model and each robust strategy on the HANS dataset (McCoy et al., 2019) that diagnose each of the three heuristics (the Lexical Overlap Heuristic, the Subsequence Heuristic, and the Constituent Heuristic).

[11] we use inputs which replace all tokens with pad token except for [SEP] and [CLS] as baseline of the attribution method.

[12] There are 407 premise and hypothesis pairs which contain word "*zero*" with a REFUTES label, while 122 pairs with a SUPPORTS label.

[13] https://github.com/chrisc36/debias

Model	Input text with label as hypothesis	Predicted	Gold-Std.
FEVER	**Original :** [CLS] rate current essay a zero [SEP] i want to rate a book . [SEP] (0.140) **Variation:** [CLS] rate current essay a one [SEP] i want to rate a book . [SEP] (0.627) **Variation:** [CLS] rate current essay a five [SEP] i want to rate a book . [SEP] (0.758)	SearchScreeningEvent (0.203) RateBook (0.627) RateBook (0.758)	RateBook
RTE	**Original :** [CLS] for the current saga i rate 2 of 6 stars [SEP] i want to rate a book . [SEP] (0.001) **Variation:** [CLS] for the current novel i rate 2 of 6 stars [SEP] i want to rate a book . [SEP] (0.925) **Variation:** [CLS] for the current essay i rate 2 of 6 stars [SEP] i want to rate a book . [SEP] (0.029)	AddToPlaylist (0.001) RateBook (0.925) SearchCreativeWork (0.043)	RateBook
MNLI	**Original :** [CLS] make me a reservation in tn somewhere nearby for a party of 4 [SEP] i want to book a restaurant . [SEP] (0.012) **Variation:** [CLS] make me a reservation in tn somewhere nearby for a party of 4 [SEP] i want to book a place . [SEP] (0.918) **Variation:** [CLS] make me a reservation in tn somewhere nearby for eating [SEP] i want to book a restaurant . [SEP] (0.797)	AddToPlaylist (0.017) - BookRestaurant (0.797)	BookRestaurant

Table A.3: Examples for visualization of attribution score. Each example is followed by the model's prediction probability for entailment class. "Predict" column shows the model's predicted class with its entailment probability for the input premise text and "Gold-Std." column displays the true labels. The red color represents negative attribution score and the blue color represents positive score for entailment class. Better viewed in color.

Text with Gold-standard and Predicted labels
• Gold-standard: Computers&Internet • Prediction: Entertainment&Music (MNLI, RTE), Computers&Internet (FEVER) *Is it possible to rip the **music** from PS2 games ? No i dont think thats possible because your computer cant understand the data format your ps2 games . Ive also never heard of that being done so id have to say no .*
• Gold-standard: Education&Reference • Prediction:Family&Relationships(RTE,FEVER,MNLI) *Who or which company would do the best **family** history and genealogy research for me in Utah ? I know if you go to the Mormon Church , they can provide tons of answers about your genealogy , and probably suggest a company or person who would do the work for you .*
• Gold-standard: BookRestaurant • Prediction: RateBook (RTE,FEVER,MNLI) ***book** a bakery for lebanese on january 11th 2032*
• Gold-standard: BookRestaurant • Prediction: RateBook(RTE,FEVER,MNLI) ***book** a highly **rated** place in in in seven years at a pub*
• Gold-standard: Negative • Prediction: Positive (RTE,FEVER,MNLI) *outer-space buffs might **love** this film , but others will find its **pleasures** intermittent .*

Table A.4: Error cases of the entailment models which may rely on spurious lexical features to make prediction. Bolded tokens indicate those cue words that may mislead the NLI models.

Template	NSP	RTE	MNLI	FEVER
The movie is great/terrible.	79.7	52.5	67.5	71.7
The movie is good/bad.	78.9	52.6	75.8	78.3
The person feels good/bad.	69.3	63.5	78.3	82.9

Table A.5: Accuracy on SST-2 dev set using different templates

Dataset	Average	Std	Min	Max
RTE Dev set	69.0	2.2	66.1	70.8
Yahoo	20.6	7.4	11.2	28.6
Emotion	3.8	0.4	3.5	4.4
Situation	23.0	4.5	16.9	28.3
AGNews	31.1	15.5	9.1	46.4
SST-2	67.0	3.9	63.9	72.0
Snips	67.5	2.5	64.3	71.3

Table A.6: Results of five runs of BERT fine-tuned on RTE and tested on classification datasets

		Entailment			*Non-entailment*		
	Overall	*L*	*S*	*C*	*L*	*S*	*C*
MNLI	53.0	99.5	99.8	97.2	2.7	1.6	17.2
w/ DA	67.3	81.2	94.6	96.6	86.8	23.7	20.7
w/ RW	64.5	69.8	80.6	78.5	53.1	40.2	65.0
w/ BP	65.4	71.4	77.4	84.6	61.0	40.7	57.2

Table A.7: HANS accuracy of BERT pretrained on MNLI and different debiasing methods, broken down by the heuristic that the example is diagnostic of and by its gold label. *L* represents for Lexical Overlap Heuristic, *S* represents for Subsequence Heuristic, and *C* represents for the Constituent Heuristic.

Neural-Symbolic Commonsense Reasoner with Relation Predictors

Farhad Moghimifar[1] and **Lizhen Qu**[2] and **Yue Zhuo**[3]
Gholamreza Haffari[2] and **Mahsa Baktashmotlagh**[1]
[1]The School of ITEE, The University of Queensland, Australia
[2]Faculty of Information Technology, Monash University, Australia
[3]School of CSE, The University of New South Wales, Australia
{f.moghimifar,m.baktashmotlagh}@uq.edu.au
firstname.lastname@monash.edu, terry.zhuo@unsw.edu.au

Abstract

Commonsense reasoning aims to incorporate sets of commonsense facts, retrieved from Commonsense Knowledge Graphs (CKG), to draw conclusion about ordinary situations. The dynamic nature of commonsense knowledge postulates models capable of performing multi-hop reasoning over new situations. This feature also results in having large-scale sparse Knowledge Graphs, where such reasoning process is needed to predict relations between new events. However, existing approaches in this area are limited by considering CKGs as a limited set of facts, thus rendering them unfit for reasoning over new unseen situations and events. In this paper, we present a neural-symbolic reasoner, which is capable of reasoning over large-scale dynamic CKGs. The logic rules for reasoning over CKGs are learned during training by our model. In addition to providing interpretable explanation, the learned logic rules help to generalise prediction to newly introduced events. Experimental results on the task of link prediction on CKGs prove the effectiveness of our model by outperforming the state-of-the-art models.

1 Introduction

Commonsense reasoning refers to the ability of capitalising on commonly used knowledge by most people, and making decisions accordingly (Sap et al., 2020). This process usually involves combining multiple commonsense facts and beliefs to draw a conclusion or judgement (Lin et al., 2019). While human trivially performs such reasoning, current Artificial Intelligence models fail, mostly due to challenges of acquiring relevant knowledge and forming logical connections between them.

Recent attempts in empowering machines with the capability of commonsense reasoning are mostly centred around large-scale Commonsense Knowledge Graphs (CKG), such as ATOMIC and ConceptNet (Sap et al., 2019; Speer et al., 2017). Unlike conventional Knowledge Graphs (KG), CKGs usually contain facts about arbitrary phrases. For instance, “PersonX thanks PersonY" is connected to “ To express gratitude" via the link “ because X wanted". This non-canonicalised free-form text representation has resulted in having conceptually related nodes with different representation, which forms *large sparse* CKGs (Malaviya et al., 2020). Therefore, established reasoning models on conventional KGs perform poorly on CGKs (Yang et al., 2014; Sun et al., 2018; Dettmers et al., 2018; Minervini et al., 2020). In addition, the nature of commonsense reasoning encourages dynamic CKGs, where new sets of facts and phrases are introduced frequently. Most existing models in this realm are based on a static set of facts and phrases, which results in poor generalisation (Malaviya et al., 2020; Shang et al., 2019). Nevertheless, the inference process in existing approaches is like a *black box*, where internal behaviour of the model is hardly interpretable.

To overcome these limitations, we propose a neural-symbolic reasoning model based on backward-chaining. While traditional theorem proving algorithms (Bratko, 2001) work based on a set of predefined rules and unification over discrete symbols, we leverage a continuous relaxation of weak unification and a rule learner module. The weak unification over continuous embedding representation helps to address the challenges of unseen sparsity of CKGs. The rule learner module, in addition to providing interpretability, is used to generalise prediction to unseen data points to mitigate the problem of *large-scale dynamic* CKGs. The experiments on the task of link prediction confirm the superiority of our model, by a margin of up to 22 points, over the state-of-the-art models.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 797–802
August 1–6, 2021. ©2021 Association for Computational Linguistics

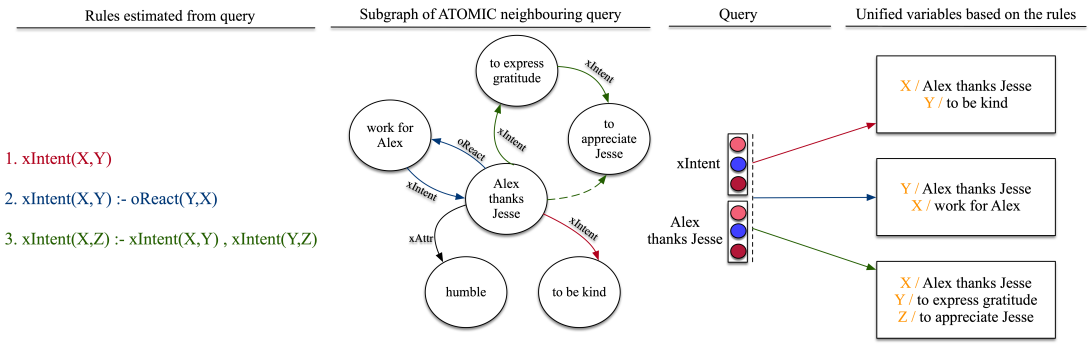


Figure 1: A visual representation of rules and new relations estimated by our model for a sample query, *xIntent(Alex thanks Jesse, ?)*. Based on the subject of the query, a subgraph of ATOMIC is retrieved for the reasoning process (middle). Sets of rules estimated from relation of the query is generated using our proposed rule creation module (left). Based on our reasoning model, the answers to query are predicted by unification module (right).

2 Related Works

Recent approaches in knowledge base completion task have mostly relied on a graph and entity-relation embedding methods (Yang et al., 2014; Dettmers et al., 2018). In these approaches, entities and relations are embedded in a complex space, and using a scoring function plausibility of a triple is estimated (Bordes et al., 2013; Trouillon et al., 2016; Sun et al., 2018). In addition to node embedding, graph embedding methods have also been used to capture the structural complexity of knowledge bases (Schlichtkrull et al., 2018; Shang et al., 2019). Language generative models also have been applied on knowledge bases in order to use the rich information of pre-trained models to address CKG completion task (Bosselut et al., 2019; Moghimifar et al., 2020). Malaviya et al. (2020) proposed a method based on using language models and graph networks to solve the problem of the sparsity of CKGs, by taking structural and contextual characteristics of CKGs into account. However, the aforementioned models are highly dependant on training on a set of static entities, and fail to perform when new triples are presented.

3 Our Approach

A CKG $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges in $\mathcal{G}$, consists of triples in form of $r(h, t)$, where $h, t \in \mathcal{N}$ are referred to as the head and the tail of the triple, and $r \in \mathcal{E}$ denotes their relation. The goal of the CKG completion task is to estimate probable t given a query $q = r(h, ?)$. As the target node may not pose a direct link to h via r, this task requires a model capable of multi-step reasoning.

Given a query $r_q(h_q, ?)$, we try to identify an implication rule and apply it to prove $r_q(h_q, t)$ for a target entity or event t. A rule $\mathcal{R}$ takes the form of $r_q(X, Z) :- r_0(X, Y_0), ..., r_k(Y_{k-1}, Z)$, where capitalised letters denote variables, $r_q(X, Z)$ is the rule head, and the rule body is a conjunction of atoms. We apply such a rule by unifying atoms with triples in the given CKG to obtain $r_q(h_q, t_k) :- r_0(h_0, t_0), r_1(t_0, t_1), ..., r_k(t_{k-1}, t_k)$, which entails $r_q(h_q, t_k)$. Since semantically equivalent/similar events or entities in a CKG often have different surface forms, we consider weak unification of an atom with a triple instead of only considering exact match of two atoms, a weak unification operator (Sessa, 2002) unifies two different symbols by measuring the similarity of their representations.

Given a query, we do not know the target rule in advance. As shown in the example in Fig. 1, we successively create a new rule by appending the body of the previous rule with an atom in the form of $r(t_{k-1}, X)$. Whenever such a new atom is added, we query the CKG to find triples as candidates of unification. This step enables reasoning on large scale KBs. In contrast, the prior works (Minervini et al., 2020; Ren and Leskovec, 2020) require comparison with each node in a CKG. After applying the weak unification operator to each of the triples, we find top$k$ most similar nodes and use each of the entity/event in the place of X to create a new atom for a new rule. The process is repeated until the maximal rule length is reached.

The above mentioned reasoning process is delivered by a a neural-symbolic reasoner. It consists of a query module, a weak unification operator, and a rule creation module.

Dataset	#Nodes	#Edges	Avg. In-degree	Density	Unseen Nodes	Unseen Edges	#Relations
ATOMIC	382823	785952	2.25	1.6e-5	38.36%	27.91%	9
ConceptNet-100k	80994	102400	1.25	9.0e-6	11%	8%	34

Table 1: Statistics on ATOMIC and ConceptNet-100k. Unseen Nodes is the ratio of the nodes in test set that are not in train set to all of the nodes in test set. Unseen edges is the ratio of edges where either the head or tail nodes are not in train set to the number of all edges in test set.

Query Given a rule with a rightmost atom $r_{k-1}(t_{k-1}, X)$ in the rule body, we send the representation of t_{k-1} as query to the given CKG to retrieve unification candidates. A node in a CKG is a word sequence. To support comparison of nodes w.r.t. their semantic similarities, we encode queries and nodes in a CKG with a pre-trained BERT (Devlin et al., 2019) into embeddings. To this end, a node is converted into $[CLS]+node+[SEP]$, and fed into the model, and we use the representation of $[CLS]$ token from the last layer of BERT as representation of node $node$. We apply FAISS[1] (Johnson et al., 2019) to index embeddings of an CKG, because it supports fast retrieval of k nearest neighbours of a dense vector. For each node v in the top$k$ list, we collect a set of triples $\mathcal{C}(v)$, which are all triples having v as the head in the CKG. As a result, we have k such sets and form a candidate set $\mathcal{C}$ by taking the union of them.

Weak Unification From a candidate set $\mathcal{C}$ we identify top$k$ most relevant triples to unify $r_{k-1}(t_{k-1}, X)$. First, we formulate a set of hypotheses $\mathcal{H}$ by replacing X with possible tails. In practice, we use all tails of the triples in C. Furthermore, we construct a bipartite graph between $\mathcal{C}$ and $\mathcal{H}$, in which an edge denotes the unification between a triple from $\mathcal{C}$ and another from $\mathcal{H}$. We measure unification scores by using cosine similarity and obtain an similarity matrix $\mathbf{U} \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{H}|}$. The final unification score of candidate triple i is computed by $\max_j \mathbf{U}_{ij}$. We keep only top$k$ highest scored candidate triples.

Rule creation Given the top$k$ highest scored candidate triples and a rule $\mathcal{R}_k$ with a rightmost atom $r_{k-1}(t_{k-1}, X)$, we create a new rule based on $\mathcal{R}_k$ for each triple k by substituting it for $r(t_{k-1}, X)$ and append another atom $r_k(t_k, X)$. The relation r_k is estimated by a relation predictor $\boldsymbol{f_\theta}(r_{k-1}, k)$, where both r_{k-1} and the current step k are mapped to the corresponding embeddings.

$$P_{\boldsymbol{\theta_f}}(r_k|r_{k-1}, k) = \sigma(\boldsymbol{f_\theta}([r_{k-1}; k]).\mathbf{W} + b) \quad (1)$$

[1]https://github.com/facebookresearch/faiss

where $\boldsymbol{\theta_f} := \{\mathbf{W}, b\}$ contains the Rule creation module's parameters, and σ is the sigmoid function. The relation predictor aims to generalise relation co-occurrence patterns in rules. We implement it by using a neural networks with two blocks of hidden layers, followed by a softmax layer. Each block is composed of a linear layer and a ReLU layer.

Given a query $r_q(h_q, ?)$, we initialise the first rule as $r_q(h_q, X)$. After reaching the pre-defined maximal rule length, we consider the score of a rule after unification as the lowest unification score associated with the rule, following (Sessa, 2002). We rank all rules by their scores and select the tails in the rule heads of the top$k$ highest scored rules as the results.

Another benefit of our reasoner is that humans can easily collect evidences to interpret reasoning results. The model can yield the rules and unified triples in a human-friendly format, which are generated at each step. In contrast, prior work (Malaviya et al., 2020) on commonsense reasoners produces only hard-to-understand distributed representations in intermediate steps.

Training We convert all the triples in $\mathcal{G}$ into a set of queries ($\mathcal{Q} = \{r_1(h_1, ?), r_2(h_2, ?), \ldots, r_n(h_n, ?), \}$), where each query of $r_i(h_i, ?)(i < n)$ is associated with a set of gold answers $\mathcal{T}_i = \{q_{i_1}, q_{i_2}, \ldots, q_{i_m}\}$. The goal of training our model is to learn the embedding representations by minimising a cross-entropy loss function ($\mathcal{L}_\theta$) on final scores associated with each estimated predictions and the set of gold answer:

$$\mathcal{L}_{\boldsymbol{\theta}} = -\sum_{q_{p_k}\in\mathcal{T}} \log(Pr(q_{p_k}|\mathcal{G}; \boldsymbol{\theta})) - \sum_{q_{p_k}\notin\mathcal{T}} \log(1 - Pr(Pr(q_{p_k}|\mathcal{G}; \boldsymbol{\theta}))) \quad (2)$$

where $\boldsymbol{\theta}$ denote all the parameters of our model. The relation predication module of our model is also trained by minimising loss in equation 2, where the relation embeddings are decoded by

	ConceptNet-100k				ATOMIC			
Model	MRR	HITS@1	HITS@3	HITS@10	MRR	HITS@1	HITS@3	HITS@10
DistMult	8.68	5.38	9.33	15.23	11.49	9.16	11.83	16.3
ComplEx	10.33	6.51	11.24	17.31	12.96	10.65	13.9	17.08
ConvE	16.55	10.19	18.79	28.08	9.04	7.05	9.42	12.74
RotatE	19.89	14.45	25.32	37.56	10.61	8.56	10.76	14.98
Malaviya et al.	43.60	39.33	49.41	66.58	23.43	20.54	24.1	27.43
Ours	**65.72**	**57.49**	**61.7**	**71.46**	**46.41**	**43.31**	**45.94**	**47.24**

Table 2: Results on CKG completion task, on ConceptNet-100K and ATOMIC.

alignment of the associated embedding and nearest predicate representation.

4 Experiments

To evaluate the performance of our model [2] in the task of CKG completion, in this section, we report the results of our model in comparison with the baselines.

Evaluation Metrics: Following previous works on Knowledge Base completion (Dettmers et al., 2018; Malaviya et al., 2020), we report the results of HITS and Mean Reciprocal Rank. Similar to Dettmers et al. (2018), when computing the scores for a gold target entity, we filter out all remaining valid entities. Furthermore, for each triple (h, r, t), the score is the average of scores measured from $(h, r, ?)$ and $(t, r^{-1}, ?)$.

Baselines For comparison, we report the performance of state-of-the-art models in CKG and KB completion. We compare our model to DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2018), RotatE (Sun et al., 2018), and Malaviya (Malaviya et al., 2020). The first four models are high performance models in conventional KB completion, whereas the latter is proposed for CKG completion.

4.1 Datasets

ATOMIC [3] is a CKG consisting of commonsense facts in form of triples, based on *if-then* relations (Sap et al., 2019). This dataset consists of more 877K facts, and more than 300K entities.

ConceptNet-100K [4] is a subset of ConceptNet 5 (Speer et al., 2017), containing Open Mind Common Sense (OMCS) entries, introduced by (Li et al., 2016). This dataset contains general commonsense facts in form of triples.

In order to evaluate the performance of the models in dynamic CKG completion, we choose a subset of the test set of ATOMIC and ConceptNet-100K, where for any (h, r, t) either h or t is not seen by the model in the train set. Statistics on ATOMIC and ConceptNet-100k are provided in table 1. To train our model, each triple in form $r(h, t)$ in train set was also converted to $r^{-1}(t, h)$, to account for reverse relations as well. We have used the embedding size of 1024 for both node and relation embedding layer. To embed the nodes in CKGs, we have fine-tuned uncased BERT-Large (Devlin et al., 2019) for the objective of masked language model. For this purpose, a node is converted into $[CLS] + n_i + [SEP]$ and fed into BERT. The representation of the token $[CLS]$ from the last layer of BERT is then used as node n_i embedded representation. We used the maximum sequence of 128, and batch size of 64. Our relation predication module consists of two Linear layer. For all non-linearities in our model we have used ReLU. For optimisation purpose, SGD has been used, with staring learning rate of $10e-4$, and decay rate of 0.9, if the loss of development set does not decrease after each epoch. We set the maximum depth of three for reasoning process. We have trained the model for 200 epochs. Followed by Malaviya et al. (2020), we have trained all the baselines for 200 epochs. During training the models were evaluated on development set, every 10 and 30 epochs, for ConceptNet-100K and ATOMIC, respectively. The checkpoint with the highest MRR was then selected for testing.

4.2 Results

Table 2 summarises the results of the conducted experiment on ConceptNet-100K and ATOMIC. On ConceptNet-100K our proposed model outperforms the baselines by up to 22 points on MRR. The gap between our model and the second best model decrease as we move from HITS@1 to HITS@10.

[2] Code available at https://github.com/farhadmfar/commonsense_reasoner

[3] https://homes.cs.washington.edu/ msap/atomic/

[4] https://ttic.uchicago.edu/ kgimpel/commonsense.html

ATOMIC
xIntent(X,Y):-xIntent(X,Z),xIntent(Z,Y)
xNeed(X,Y):-xReact(Y,X)
xIntent(X,Y):-oWant(Y,X)
ConceptNet-100K
causes(X,Y):-causes(X,Z),causes(Z,Y)
isa(X,Y):-partof(X,Z),isa(Z,Y)
relatedto(X,Y):-relatedto(X,Z),relatedto(Z,Y)

Table 3: Examples of rules learned by our proposed relation prediction module.

This suggested that on contrary to the baselines our model performs better in estimating the probability of query with higher accuracy. On ATOMIC our model achieves a MRR of 46.41, which is 23 points higher than the second best model. As it can be seen from table 2, comparison of performance of different models on ConceptNet-100K and ATOMIC shows a noticeable drop in performance for models which rely on structural information of CKGs. This observation suggests that larger and sparser (lowest density) CKG are more challenging to reason over.

Table 3 provides examples of generated rules by our model on ATOMIC and ConceptNet-100k. On ATOMIC, the first rule is based on transition, and the second and third rules are inverse rules. Similarly, on ConceptNet-100K the first and third rules are transitive, and the second rule is a compositional rule. All provided rules are diverse and meaningful, and can be used for explaining the inference process of our model. For instance, consider a query of *xIntent(Alex drives Jesse there, ?)*. Based on first rule from Table 3, `X` is unified by *Alex drives Jesse there*, and `Z` is unified by *Alex helps Jesse* (from triples of ATOMIC). Then, the query is updated to *xIntent(Alex helps Jesse, ?)* and `Y` is unified by *to be of assistance* (from triples of ATOMIC), hence the answer to query. The path generated by this example is *Alex drives Jesse there* $\xrightarrow{\text{xIntent}}$ *Alex helps Jesse* $\xrightarrow{\text{xIntent}}$ *to be of assistance*. Therefore, two nodes are connected via a new link: *Alex drives Jesse there* $\xrightarrow{\text{xIntent}}$ *to be of assistance*.

Consider the following query from ConceptNet-100K, $HasProperty(novel, ?)$. Based on the relation of the query, our rule creator module can estimate the following rule: According to this rule,

```
HasProperty(X,Y):-IsA(X,Z),HasProperty(Z,Y)
```

`X` is unified by *novel*, and `Z` is unified by *book* (from triples of ConceptNet-100K). Then, the query is updated to *HasProperty(book, ?)* and `Y` is unified by *expensive* (from triples of ConceptNet-100K), resulting the answer to the query, by generating the following path: *novel* $\xrightarrow{\text{IsA}}$ *book* $\xrightarrow{\text{HasProperty}}$ *expensive*, hence *novel* $\xrightarrow{\text{HasProperty}}$ *expensive*.

5 Conclusion

In this work, we propose a neural-symbolic reasoning model over Commonsense Knowledge Graphs (CKGs). Our proposed model leverages a relation prediction module, which provides capability of multi-step reasoning. This ability, alongside weak unification, helps generalising our model to large-scale unseen data. We showed that our model yields state-of-the-art results when applied to large-scale sparse CKGs, and the inference step is interpretable.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ivan Bratko. 2001. *Prolog programming for artificial intelligence*. Pearson education.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph

networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2822–2832.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. In *AAAI*.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. 2020. Learning reasoning strategies in end-to-end differentiable proving. In *International Conference on Machine Learning*.

Farhad Moghimifar, Lizhen Qu, Yue Zhuo, Mahsa Baktashmotlagh, and Gholamreza Haffari. 2020. Cosmo: Conditional seq2seq-based mixture model for zero-shot commonsense question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. *Advances in Neural Information Processing Systems*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Introductory tutorial: Commonsense reasoning for natural language processing. *Association for Computational Linguistics (ACL 2020): Tutorial Abstracts*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*.

Maria I Sessa. 2002. Approximate reasoning by similarity-based sld resolution. *Theoretical computer science*, 275(1-2):389–426.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

What Motivates You? Benchmarking Automatic Detection of Basic Needs from Short Posts

Sanja Štajner[1], **Seren Yenikent**[1], **Bilal Ghanem**[2], **Marc Franco-Salvador**[2]
[1]Symanto Research, Nuremberg, Germany
[2]Symanto Research, Valencia, Spain
{sanja.stajner, seren.yenikent, marc.franco}@symanto.com
bilalhgm@gmail.com

Abstract

According to the self-determination theory, the levels of satisfaction of three basic needs (competence, autonomy and relatedness) have implications on people's everyday life and career. We benchmark the novel task of automatically detecting those needs on short posts in English, by modelling it as a ternary classification task, and as three binary classification tasks. A detailed manual analysis shows that the latter has advantages in the real-world scenario, and that our best models achieve similar performances as a trained human annotator.

1 Introduction

Motivation is one of the most crucial aspects of human behaviour with implications ranging from daily life to career and educational contexts. Self-determination theory (SDT) provides a meta-framework for understanding the broad, as well as specific, nutriments of the function and application of the concept of motivation (Deci and Ryan, 2000; Ryan and Deci, 2017a).

SDT differs from the other motivational theories from the psychology literature in two substantial aspects (Ryan and Deci, 2000; Rigby and Ryan, 2018): (1) Unlike the drive theories that explain motivation as a function of its deficit (e.g. people are motivated by success to compensate its deficit), SDT focuses on growth and constructivism (e.g. people are naturally and universally motivated by success), thus giving the theory a more realistic understanding of the human behaviour, and making it applicable to wider contexts; and (2) Due to the applicability advantage, SDT is based on strong behavioural evidence and is thus not only a well-validated model but also sustainable and actionable.

The SDT framework is supported by a body of cross-cultural studies strengthening the universality of the theory. Studies conducted in diverse countries showed that the basic needs are essentially represented across cultures (Chen et al., 2015; Jang et al., 2009). Although universal, the SDT framework is also able to point out the impact of sociocultural environment on the variations of basic needs in different cultures. For example, a study conducted in 11 countries showed that the need for competence was more linked to school performance in Eastern cultures than in the West (Nalipay et al., 2019).

One of the central pillars of SDT are three basic psychological needs that drive the initiation of a behaviour and the maintenance of motivation:

- **Autonomy**: the basic need to be the owner and controller of one's decisions and behaviours.
- **Competence**: the basic need to feel competent, effective and master-like.
- **Relatedness**: the basic need to belong, bond and connect with others.

According to SDT, those three needs are universal and their importance does not change across individuals and situations. However, different contexts and time periods would require different support and resources for the maintenance of the motivations. For instance, cultivating autonomy need in students creates more engagement and willingness, thus leading to higher academic performance, lower dropouts, and more self-esteem in the long run (Ryan and Deci, 2020). Similarly, the SDT framework is used to increase levels of employee satisfaction and engagement, supportive leadership and parenting skills, healthier relationships, satisfactory consumer experience and better designed digital media and well-being tools (Slemp et al., 2018; Rigby and Ryan, 2018; Ryan and Deci, 2017b; Knee et al., 2002; Gilal et al., 2019; Peters et al., 2020; Peng et al., 2012).

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 803–810
August 1–6, 2021. ©2021 Association for Computational Linguistics

Need	Post
Autonomy	Just treated myself to a Roland TB-3. Should arrive this evening. #excited
Autonomy	One thing's for sure, I will not let you ruin my dreams, HIV. #determined
Competence	What an achievement. Finally getting some credit. #Fury #SPOTY
Competence	I fell asleep with socks on... I disgust myself.
Relatedness	I'm so lucky to have my best friend and boyfriend rolled into one! #soppy #proud
Relatedness	You know what I feels like to be #ALONE in this cold world?

Table 1: Annotated examples from the dataset (either satisfied or unsatisfied need).

Traditionally, basic motivations are assessed via questionnaires which provide intensity-based scores for each dimension. The scores represent the degree to which that particular dimension is satisfied (Deci and Ryan, 2000). Although these questionnaires were developed and validated via laboratory and field studies which provide a strong empirical basis, they could suffer from biases commonly observed in questionnaire respondents such as social desirability bias (Krumpal, 2011) and the reference-group effect (Heine et al., 2002). The basic motivations can also be revealed in a more implicit way, by collecting subjects' narratives while showing them pictures and images (Murray, 1943; McClelland, 1979). Although being more expensive and time-consuming, as it requires the inclusion of trained assessors, this method shows that implicit motivations can be assessed from texts. A few studies attempted at automatic detection of basic motivations on the basis of their linguistic aspects from such naratives (Pennebaker and King, 1999; Johannssen and Biemann, 2019).

To the best of our knowledge, our study is the first that attempts to automatically detect the three basic needs from short posts. In this study, we:

- Benchmark the task of automatic detection of basic needs from English Twitter data using several architectures on an already existing manually annotated dataset.
- Provide a manual analysis which shed light on the complexity of the task and its usability.
- Discuss the limitations of the existing dataset, and suggest better annotation strategies.

2 Dataset

For our experiments, we used the first two layers of the Basic Psychological Needs Corpus (Alharthi et al., 2017), which is publicly available.[1] The corpus contains Twitter posts annotated with five layers of annotation as the intention was to provide freely available multilayered annotated corpus for a wide range of applications (Alharthi et al., 2017). The manual annotation was performed by three annotators in three stages, encompassing thorough training sessions and detailed annotation guidelines, one round of collectively labelling tweets, one round of independently labelling the same posts for calculating inter-annotator agreement (IAA), and the final round of independently labelling the rest of the posts. The average pairwise agreement and the Fleiss Kappa (κ) were 90% and 0.815 for whether or not the post contains enough content for assigning one of the three basic needs (autonomy, relatedness, or competence), and 89% and 0.819, respectively, for the assigned label (Alharthi et al., 2017).

The final dataset with manual annotations of basic needs was already pre-filtered for non-emotional posts and those that do not contain enough signal (Alharthi et al., 2017). It contains 6334 posts with the following distribution of the labels: 1229 posts labelled with *competence*, 1771 with *autonomy*, and 3334 with *relatedness* label. In our experiments, we used this dataset and only the labels of the second layer of annotation (basic needs). Several examples are given in Table 1. Here is important to note that the original dataset also contains, in the third layer, the annotation for the satisfaction level (satisfied, dissatisfied, neutral) of the assigned basic need. We acknowledge that the combination of the basic needs and their level of satisfaction are often used together, e.g. as indicators of person's well-being (Deci and Ryan, 2011), violence and conflict possibility (Christie, 1997), stress and coping (Ntoumanis et al., 2008; Weinstein and Ryan, 2011). However, we opted for discarding these additional labels for three reasons: (1) because the inter-annotator agreement was significantly lower for this annotation layer

[1] We obtained the original dataset directly from the authors.

(the average pairwise agreement was 75% and the κ was 0.640); (2) so that we do not increase the total number of classes (to nine instead of three) and therefore significantly lower the number of instances in each class; (3) because this task appears similar to the task of assigning the sentiment polarity of the post (Alharthi et al., 2017), and therefore might be modelled with various other datasets.

3 Experimental Setup

3.1 Preprocessing

The instances were already cleaned in the original dataset by removing all usernames (@username) and URLs, while preserving emoticons, punctuation marks, social acronyms and abbreviations, which might contain psycholinguistic signals (Alharthi et al., 2017). Furthermore, the dataset does not contain any duplicated instances, tweets with less than three words, or tweets with more than three hashtags (Alharthi et al., 2017). We noticed that for this particular task, the hashtags may help the models, e.g. *#proud* usually signalizes *competence*, *#relationship* signalizes *relatedness*. To better assess how well the models would perform on a different type of texts, we experimented with two versions of the dataset: WITHOUT HASHTAGS and WITH HASHTAGS.

3.2 Data Splits

We randomly choose 15% of the instances for testing, and then 15% from the rest of the data for development, while maintaining the class ratio (Table 2). During our experiments, we found that applying upsampling on the minority classes (*competence* and *autonomy*) slightly improved the performances of some models, and had no change on others. Thus, we only report the results obtained by using upsampling.

3.3 Task Definition

We approached the problem of detecting basic needs with two different scenarios: (1) as a ternary classification problem (assigning one of the three possible basic needs to each post), and (2) as three binary classification tasks (for each basic need, assigning either *yes* or *no* label). The ternary classification is a more natural choice for this particular dataset, as all instances were annotated with only one of the three basic needs. However, according to the SDT, each person have at all times the all three needs just with different intensities and satisfaction levels (Section 1). It is thus reasonable to assume that some posts will also contain signals of multiple basic needs. Therefore, we also performed three binary tasks which would allow us to model each basic need separately. By using three binary classifiers instead of one ternary, posts could be automatically labelled with none of, or any combination of, basic needs.

Need	ORIGINAL			REPORTED		
	TRAIN	DEV	TEST	TRAIN	DEV	TEST
Autonomy	1248	228	290	2416	404	290
Competence	868	168	204	2416	404	204
Relatedness	2416	404	508	2416	404	508

Table 2: Dataset splits.

3.4 Evaluation Metrics

For both types of classification tasks (binary and ternary), we used the per-class precision, recall, and F_1-score, and the macro-averaged F_1-score for evaluating the performances of the models.

3.5 Architectures

In order to assess the importance of both lexical and semantic aspects of texts, we tested various approaches that use different text representations:

- **BOW**: word unigrams and bigrams model with the TF-IDF weighting scheme (Salton and Buckley, 1988) using a Support Vector Machines (Chang and Lin, 2011) classifier with a linear kernel.[2,3]
- **Char-CNN**: a Convolutional Neural Networks (CNN) architecture similar to the one proposed in (Zhang et al., 2015) but using a trainable character embedding layer as input.
- **BiLSTM**: a bidirectional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) neural network that uses FastText word embeddings (Bojanowski et al., 2017) to represent texts. The BiLSTM hidden states are fed to an attention layer (Yang et al., 2016), and then the attention output is processed with a fully connected layer. As an output, a softmax layer is used to obtain the final classification.

[2]We also explored logistic regression, random forest, Naive Bayes, and support vector machines, with different kernels, during the prototyping phase.

[3]Character n-grams were also tested but as they did not lead to better performances, we do not report their results.

Model	WITHOUT HASHTAGS										WITH HASHTAGS									
	Autonomy			Competence			Relatedness			F_1	Autonomy			Competence			Relatedness			F_1
	P	R	F_1	P	R	F_1	P	R	F_1	(macro)	P	R	F_1	P	R	F_1	P	R	F_1	(macro)
BiLSTM	.54	.54	.54	.48	.45	.47	.74	.76	.75	.59	.65	.47	.55	.47	.63	.54	.80	.82	.81	.63
Char-CNN	.50	.64	.56	.58	.24	.34	.73	.79	.76	.55	.61	.61	.61	.54	.54	.54	.82	.81	.82	.66
BOW	.61	.49	.54	.49	.54	.52	.76	.82	.79	.62	.67	.57	.62	.57	.54	.55	.79	.87	.83	.67
BERT	.62	.54	.58	.55	.61	.58	.84	.87	**.86**	.67	.70	.69	**.69**	.73	.52	.61	.83	.93	**.88**	.72
BERT+BiLSTM	.62	.60	**.61**	.56	.66	**.60**	.87	.82	.85	**.69**	.71	.61	.66	.69	.62	**.65**	.83	.93	**.88**	**.73**
Trained human	.78	.70	**.74**	.69	.88	**.77**	.88	.72	.79	**.77**	.78	.70	**.74**	.69	.88	**.77**	.88	.72	.79	**.77**
BERT+BiLSTM	.78	.70	**.74**	.79	.65	.71	.73	.93	**.81**	.75	.73	.75	**.74**	.81	.62	.70	.77	.93	**.84**	.76

Table 3: Results of the ternary classification task. The last two rows present the results on a subset of the test set that was annotated by a trained human annotator and contains 40 instances of each class.

- **BERT**: the neural language model, well-known for providing text representations that show leading performances on several natural language processing benchmarks (Devlin et al., 2019). We fine-tune BERT and use its hidden representation of the special [CLS] token to represent the full input text and feed it to a softmax output layer.

- **BERT+BiLSTM**: this model combines the previous two approaches. Instead of FastText word representation, the fine-tuned BERT embeddings are post-processed by the BiLSTM architecture defined above. We observed that such architectures help BERT to adapt to the target task and obtain better classification results in scenarios with small training datasets.

4 Results and Discussion

4.1 Ternary Classification

All models performed noticeably better on the original than on the cleaned dataset, thus supporting our hypothesis that the presence of the hashtags leads to better model performances (Table 3). As expected, the models that are based on transfer learning (BERT and BERT+BiLSTM) performed best. Interestingly, the non-neural model (BOW) outperformed the BiLSTM and Char-CNN models on the *competence* class using the cleaned dataset (F_1-score of 0.52 against 0.47 and 0.34, respectively).

In all models, most misclassifications were observed between the *competence* and *autonomy* classes. A possible reason for this might lie in the SDT theory, as autonomy and competence are self-originated needs, whereas relatedness includes both self and others (Vansteenkiste et al., 2020). This might lead to theme/topic overlaps between autonomy and competence due to the self-focus, while relatedness might be easier to distinguish due to including self and the others.

4.2 Human Performance and Error Analysis

To assess the expected performance ceiling, we hired a psychologist, well-versioned in SDT, provided the annotation guidelines with several examples, and asked to annotate randomly selected 150 instances from the cleaned test set (50 from each class). The annotator was allowed to assign as many classes as needed to each post.

Our guidelines were based on a thorough review of psychology research by Ryan and Deci (2020, 2017a,b, 2000) who studied observable behavioural outcomes. We selected the following cues for each basic need:

- Autonomy: focus of initiative, ownership of self-actions, feelings of restriction by any type of external control.

- Competence: focus on behaviours associated with mastery, achievements, success, and growth (both positive and negative), search for personal or contextual challenges, well-structured environments, and positive feedback.

- Relatedness: focus on spending and appreciating time with significant others, search for community and connection, sense of nurturing and caring for others.

The annotator assigned two classes in 14 cases (9.3%). Some of those were the cases in which our best system (BERT+BiLSTM) made 'wrong' prediction, which turned out to be the same as one of the classes assigned by the human annotator

Gold	Predicted	Post
Relatedness	Autonomy	Wishing I was home this Christmas, maybe next year #homesick #holidays
Relatedness	Competence	I work with an amazing team. They work so hard and are so dedicated. Truly a top comms team #proud

Table 4: Examples of penalized predictions which actually caught the secondary signal. For those examples, the human annotator assigned both classes (the gold and the predicted one).

Task	Yes			No			F_1
	P	R	F_1	P	R	F_1	(macro)
Autonomy	.74	.48	.58	.81	.93	.87	.73
Competence	.69	.63	.66	.91	.93	.92	.79
Relatedness	.85	.92	.88	.91	.83	.87	.87

Table 5: Results of the binary classification tasks on the datasets WITH HASHTAGS.

Task	Yes			No			F_1
	P	R	F_1	P	R	F_1	(macro)
Autonomy	.61	.49	.54	.81	.87	.84	.69
Competence	.55	.64	.59	.90	.87	.88	.74
Relatedness	.83	.86	.85	.85	.82	.84	.84

Table 6: Results of the binary classification tasks on the datasets WITHOUT HASHTAGS.

(Table 4). Therefore, we took 120 instances for which the human annotator assigned only one class, and additionally ran our best model on that portion of the test set, to fairly compare its performance with the human performance (the last two rows in Table 3).

4.3 Binary Classifications

The results of the best performing architecture (BERT+BiLSTM) on the binary tasks using the datasets WITH HASHTAGS and WITHOUT HASHTAGS are presented in Tables 5 and 6.

To assess the performance of those systems in the real-world scenario, we took 100 random new tweets and ran all three models on them. At the same time, we asked the psychologist to annotate each post (without showing the obtained automatic predictions) by assigning one of the three labels (*no*, *low*, *high*) for each basic need. For example, *"@matchbox_sized Wait, you've seen it already? Thought it aired on Sunday nights?"* was annotated as *low* for *relatedness*, *high* for *autonomy*, and *no* for *competence*. For the same example, the three best binary models assigned the following probabilities to each of the corresponding classes: $p(autonomy) = 0.88$, $p(relatedness) = 0.70$, and $p(competence) = 0.30$.

We further investigated whether or not the class probabilities obtained by the binary models were related to the labels assigned by the annotator. On those 100 examples, we found that the manually assigned label *no* corresponds to the $p(yes) \in [0, 0.5)$ (obtained by the models) in 90% of the cases, the manually assigned label *low* to the $p(yes) \in [0.5, 0.75)$ (obtained by the models) in 100% of the cases, and manually assigned label *high* to the $p(yes) \in [0.75, 1]$ (obtained by the models) in 98% of the cases. These findings indicate that it might be possible to use the binary models in a more general setup, i.e. on the posts which are not pre-filtered for containing emotions or needs signals, and on posts that reflect more than one need. Furthermore, it seems that those models could capture the intensity of the signals.

5 Conclusions

In this study, we benchmarked the automatic detection of basic motivations on short (Twitter) posts in English, framing the problem as a ternary classification task, as well as three binary classification tasks. On the ternary classification task, our BERT+BiLSTM model performed almost equally well as a trained human annotator.

We showed that modelling this problem as three binary classification tasks, instead of modelling it as one ternary classification task, allows for better applicability of the models. The proposed setup with three binary models assigns none of the basic motivations to those posts without any signal (all three models assign a *no* class), and multiple basic motivations to those posts with signals from multiple motivations (more than one model assigns a *yes* class), achieving a high agreement with the human annotator. We also found a high association between the class probabilities of the binary models and the human-perceived motivation intensities.

6 Ethics/Impact Statement

6.1 Intended Use

The goal of our experiments was to investigate if there is a possibility to automatically detect basic needs from short posts, and to benchmark this novel NLP task. As we do not have any demographic information in the dataset used, and we did not thoroughly investigate performances of our models on different text types, demographic groups, and in different contexts, we do not encourage the use of these particular models in real-world applications. Instead, the contribution of our study lies in setting the ground for future models of automatic detection of basic needs from short texts, by benchmarking the task with various machine learning architectures on a specific dataset, experimenting with both ternary and binary setups, providing performance ceiling estimation via human annotations, and discussing the usability of presented approaches. Our study thus provides the foundations for future models which, if trained on carefully sampled data (representative data with strict bias control), have the potential to speed up and provide additional quality checks for traditional questionnaire-based basic needs estimation procedures, which are already widely used for: (1) providing supportive information about the user in organizational contexts such as leadership style and team building processes (Rigby and Ryan, 2018); and (2) prompting learner perspectives in educational contexts such as designing motivation-supportive settings and activities (Schneider et al., 2018).

6.2 Failure Modes

To try to estimate how the model would perform if trained on different type of data, i.e. non-Twitter data, we evaluated models trained on posts with hashtags and models trained on the same posts but after removing all hashtags. However, it is not certain how would the reported models perform on different types of data, neither whether training models with different data sources would lead to similar results or not. On the used Twitter datasets, we found most misclassifications between *autonomy* and *competence* classes.

6.3 Biases

Given that we do not have any demographic information about the authors of the posts in the used dataset, and that the dataset was prefiltered for emotional and needs signals (Alharthi et al., 2017), the presented models might suffer from various algorithmic biases. Furthermore, it is known that certain age groups or socio-economic groups are more present in Twitter than others (Tufekci, 2014; Morstatter et al., 2014), and that certain personality types are more active on particular media platforms (Goby, 2006).

6.4 Misuse Potential

Using automatic detection of basic needs in decision-making processes during hiring and placement could lead to a potential misuse and unfair decisions due to: (1) algorithmic biases and imperfections of the models; (2) giving too much weight to the estimation of basic needs instead of taking it only as one of many aspects of the employee (e.g. personality, educational background) and team work.

Basic needs could be used in combination with other psychological variables (e.g. personality) for marketing and consumer targeting purposes. Tailoring marketing materials for different personalities can be beneficial for consumers by leading them to spend their money on personality-matching items (Matz et al., 2016). However, it can also be misused by leading people to act against their best interests, e.g. by persuading them to gamble (Matz et al., 2016).

6.5 Potential Harm to Vulnerable Populations

As any other psychological modelling, when combined with demographic characteristics (e.g. age, gender, socio-economic background), machine learning models could potentially harm vulnerable groups such as immigrants or people with mental health issues. The models could potentially detect people who suffer from psychological and emotional instability, as it is highly likely that those people may be unsatisfied about their basic needs. To avoid such unintended harms, special attention should be given to carefully collecting a representative sample for any intended use (Williams et al., 2018).

References

Rajwa Alharthi, Benjamin Guthier, Camille Guertin, and Abdulmotaleb El Saddik. 2017. A dataset for psychological human needs detection from social networks. *IEEE Access*, 5:9109–9117.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Beiwen Chen, M. Vansteenkiste, W. Beyers, L. Boone, E. Deci, J. Kaap-Deeder, B. Duriez, W. Lens, Lennia Matos, Athanasios Mouratidis, R. Ryan, K. Sheldon, B. Soenens, S. Petegem, and Joke Verstuyf. 2015. Basic psychological need satisfaction, need frustration, and need strength across four cultures. *Motivation and Emotion*, 39:216–236.

Daniel J. Christie. 1997. Reducing direct and structural violence: The human needs theory. *Peace and Conflict: Journal of Peace Psychology*, 3(4):315–332.

Edward L. Deci and Richard M. Ryan. 2000. The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. *Psychological inquiry*, 4(11):227–268.

Edward L. Deci and Richard M. Ryan. 2011. Levels of analysis, regnant causes of behavior and well-being: The role of psychological needs. *Psychological Inquiry*, 22(1):17–22.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Faheem Gul Gilal, Jian Zhang, Justin Paul, and Naeem Gul Gilal. 2019. The role of self-determination theory in marketing science: An integrative review and agenda for research. *European Management Journal*, 1(37):29–44.

Valerie Priscilla Goby. 2006. Personality and online/offline choices: MBTI profiles and favored communication modes in a Singapore study. *CyberPsychology & Behavior*, 9:5–13.

Steven J. Heine, Darrin. R. Lehman, Kaiping Peng, and Joe Greenholtz. 2002. What's wrong with cross-cultural comparisons of subjective likert scales?: The reference-group effect. *Journal of Personality and Social Psychology*, 82(6):903—-918.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hyungshim Jang, Johnmarshall Reeve, Richard M. Ryan, and Ahyoung Kim. 2009. Can Self-Determination Theory Explain What Underlies the Productive, Satisfying Learning Experiences of Collectivistically Oriented Korean Students? *Journal of Educational Psychology*, 101:644–661.

Dirk Johannssen and Chris Biemann. 2019. Neural classification with attention assessment of the implicit-association test omt and prediction of subsequent academic success. In *Proceedings of the 15th Conference on Natural Language Processing, KONVENS*. German Society for Computational Linguistics & Language Technology.

C. Raymond Knee, Heather Patrick, Nathaniel A. Vietor, Aruni Nanayakkara, and Clayton Neighbors. 2002. Self-Determination as Growth Motivation in Romantic Relationships. *Personality and Social Psychology Bulletin*, 5(28):609–619.

Ivar Krumpal. 2011. Determinants of social desirability bias in sensitive surveys: a literature review. *Quality & Quantity*, 47(4).

Sandra C. Matz, Joe J. Gladstone, and David Stillwell. 2016. Money buys happiness when spending fits our personality. *Psychological science*, 27(5):715–725.

David C. McClelland. 1979. Inhibited power motivation and high blood pressure in men. *Journal of Abnormal Psychology*, 88:182–190.

Fred Morstatter, Jürgen Pfeffer, and Huan Liu. 2014. When is it biased? assessing the representativeness of twitter's streaming api. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, page 555–556, New York, NY, USA. Association for Computing Machinery.

Henry A. Murray. 1943. *Thematic Apperception Test*. Harvard University Press, Cambridge, MA.

Ma Jenina N. Nalipay, Ronnel B. King, and Yuyang Cai. 2019. Autonomy is equally important across East and West: Testing the cross-cultural universality of self-determination theory. *Journal of adolescence*, 78:67–72.

Nikos Ntoumanis, Jemma Edmunds, and Joan L. Duda. 2008. Understanding the coping process from a self-determination theory perspective. *British Journal of Health Psychology*, 14(2):249–260.

Wei Peng, Jih-Hsuan Lin, Karin A Pfeiffer, and Brian Winn. 2012. Need satisfaction supportive game features as motivational determinants: An experimental study of a self-determination theory guided exergame. *Media Psychology*, 2(15):175–196.

James W. Pennebaker and Laura A. King. 1999. Linguistic styles: Language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296–1312.

Dorian Peters, Naseem Ahmadpour, and Rafael A Calvo. 2020. Tools for wellbeing-supportive design: Features, characteristics, and prototypes. *Multimodal Technologies and Interactio*, 3(4).

C. Scott Rigby and Richard M. Ryan. 2018. Self-determination theory in human resource development: New directions and practical considerations. *Advances in Developing Human Resources*, 2(20):133–147.

Richard M. Ryan and Edward L. Deci. 2000. The darker and brighter sides of human existence: Basic psychological needs as a unifying concept. *Psychological inquiry*, 4(11):319–338.

Richard M. Ryan and Edward L. Deci. 2017a. *Self-determination theory: Basic psychological needs in motivation, development, and wellness*. Guilford Publications, New York, USA.

Richard M. Ryan and Edward L. Deci. 2017b. *Self-determination theory: Basic psychological needs in motivation, development, and wellness*. Guilford Publications.

Richard M. Ryan and Edward L. Deci. 2020. Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary Educational Psychology*, (101860).

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sascha Schneider, Steve Nebel, Maik Beege, and Günter Daniel Rey. 2018. The autonomy-enhancing effects of choice on cognitive load, motivation and learning with digital media. *Learning and Instruction*, 58:161–172.

Gavin R. Slemp, Margaret L. Kern, Kent J. Patrick, and Richard M. Ryan. 2018. Leader autonomy support in the workplace: A meta-analytic review. *Motivation and emotion*, 5(42):706–724.

Zeynep Tufekci. 2014. Big questions for social media big data: Representativeness, validity and other methodological pitfalls. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1).

Maarten Vansteenkiste, Richard M. Ryan, and Bart Soenens. 2020. Basic psychological need theory: Advancements, critical themes, and future directions. *Motivation and Emotion*, 44:1–31.

Netta Weinstein and Richard M. Ryan. 2011. A self-determination theory approach to understanding stress incursion and responses. *Stress & Health*, 27(1):4–17.

Betsy Anne Williams, Catherine F Brooks, and Yotam Shmargad. 2018. How algorithms discriminate based on data they lack: Challenges, solutions, and policy implications. *Journal of Information Policy*, 8:78–115.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *Advances in neural information processing systems*, 28:649–657.

Semantic Frame Induction using Masked Word Embeddings and Two-Step Clustering

Kosuke Yamada[1] **Ryohei Sasano**[1,2] **Koichi Takeda**[1]
[1]Graduate School of Informatics, Nagoya University, Japan
[2]RIKEN Center for Advanced Intelligence Project, Japan
yamada.kosuke@c.mbox.nagoya-u.ac.jp,
{sasano,takedasu}@i.nagoya-u.ac.jp

Abstract

Recent studies on semantic frame induction show that relatively high performance has been achieved by using clustering-based methods with contextualized word embeddings. However, there are two potential drawbacks to these methods: one is that they focus too much on the superficial information of the frame-evoking verb and the other is that they tend to divide the instances of the same verb into too many different frame clusters. To overcome these drawbacks, we propose a semantic frame induction method using masked word embeddings and two-step clustering. Through experiments on the English FrameNet data, we demonstrate that using the masked word embeddings is effective for avoiding too much reliance on the surface information of frame-evoking verbs and that two-step clustering can improve the number of resulting frame clusters for the instances of the same verb.

1 Introduction

Semantic frame induction is a task of mapping frame-evoking words, typically verbs, into semantic frames they evoke (and the collection of instances of words to be mapped into the same semantic frame forms a cluster). For example, in the case of example sentences from FrameNet (Baker et al., 1998) shown in (1) to (4) in Table 1, the goal is to group the examples into three clusters according to the frame that each verb evokes; namely, {(1)}, {(2)}, and {(3), (4)}. Unsupervised semantic frame induction methods help to automatically build high-coverage frame-semantic resources.

Recent studies have shown the usefulness of contextualized word embeddings such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) for semantic frame induction. For example, the top three methods (Arefyev et al., 2019; Anwar et al., 2019; Ribeiro et al., 2019) in Subtask-A of SemEval-2019 Task 2 (QasemiZadeh et al., 2019) perform clustering of contextualized word embeddings of frame-evoking verbs. However, these methods have two potential drawbacks.

(1)	We'll not get there before the rain comes.	(ARRIVING)
(2)	The problem continued to get worse.	(TRANSITION_TO_STATE)
(3)	You may get more money from the basic pension.	(GETTING)
(4)	We have acquired more than 100 works.	(GETTING)

Table 1: Example sentences of verbs "get" and "acquire" and frames that each verb evokes in FrameNet. (FRAME)

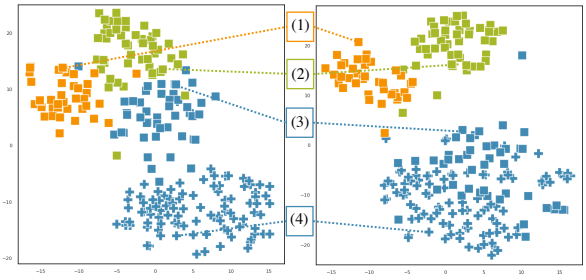


Figure 1: 2D projections of BERT embeddings of verbs (left) and masked verbs (right). Numbers in the figure correspond to numbers in Table 1, ■ and + are verbs "get" and "acquire", respectively, and each color indicates ARRIVING, TRANSITION_TO_STATE, and GETTING frame.

First, the contextualized word embeddings of the frame-evoking verbs strongly reflect the superficial information of the verbs. The left side of Figure 1 shows a 2D projection of contextualized embeddings of instances of the verbs "get" and "acquire" extracted from example sentences in FrameNet. Specifically, we extracted instances of "get" and "acquire" from FrameNet, obtained their embeddings by using a pre-trained BERT, and projected them into two dimensions by using t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008). As shown in the figure, among instances of "get", those that evoke the GETTING frame tend to be located close to instances of "acquire" that evokes the same GETTING frame. However, we can see that the difference be-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 811–816
August 1–6, 2021. ©2021 Association for Computational Linguistics

tween verbs is larger than the difference between the frames that each verb evokes.

To remedy this drawback, we propose a method that uses a masked word embedding, a contextualized embedding of a masked word. The right side of Figure 1 shows a 2D projection of masked word embeddings for instances of the verbs "get" and "acquire". The use of masks can hide the superficial information of the verbs, and consequently we can confirm that instances of verbs that evoke the same frame are located close to each other.

The second drawback is that these methods perform clustering instances across all verbs simultaneously. Such clustering may divide instances of the same verb into too many different frame clusters. For example, if there are outlier vectors that are not typical for a particular verb, they tend to form individual clusters with instances of other frames in most cases. To solve this problem, we propose a two-step clustering, which first performs clustering instances of the same verb according to their meaning and then performs further clustering across all verbs.

2 Proposed Method

The proposed semantic frame induction method uses masked word embeddings and two-step clustering. We explain these details below.

2.1 Masked Word Embedding

A masked word embedding is a contextualized embedding of a word in a text where the word is replaced with a special token indicating that it has been masked, i.e., "[MASK]" in BERT. Our method leverages masked word embeddings of frame-evoking verbs in addition to standard contextualized word embeddings of frame-evoking verbs. In this paper, we consider the following three types of contextualized word embeddings.

v_{WORD}: Standard contextualized embedding of a frame-evoking verb.

v_{MASK}: Contextualized embedding of a frame-evoking verb that is masked.

$v_{\text{W+M}}$: The weighted average of the above two, which is defined as:

$$v_{\text{W+M}} = (1-\alpha) \cdot v_{\text{WORD}} + \alpha \cdot v_{\text{MASK}}. \quad (1)$$

Here, $v_{\text{W+M}}$ is the weighted average of contextualized word embeddings with and without masking the frame-evoking verb. By properly setting the weight α using a development set, we expect to obtain embeddings that properly adjust the weight of superficial information of the target verb and information obtained from its context. $v_{\text{W+M}}$ is identical to v_{WORD} when α is set to 0 and identical to v_{MASK} when α is set to 1.

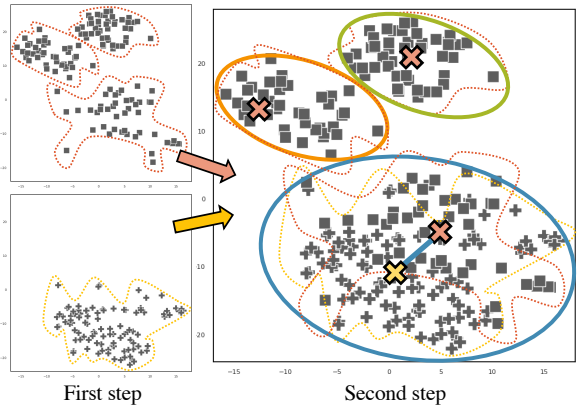


Figure 2: Flow of the two-step clustering. ■ and + denote the embeddings of "get" and "acquire", respectively.

2.2 Two-Step Clustering

In the two-step clustering, we first perform clustering instances of the same verb according to the semantic meaning and then perform further clustering across verbs. Finally, each generated cluster is regarded as an induced frame. Figure 2 shows the flow of the two-step clustering using the instances of "get" and "acquire" from FrameNet. As a result of the clustering in the first step, the instances of "get" are grouped into three clusters and the instances of "acquire" into one cluster. In the second step, one of the clusters of "get" and the cluster of "acquire" are merged. Consequently, three clusters are generated as the final clustering result. The details of each clustering are as follows.

Clustering Instances of the Same Verb The clustering in the first step aims to cluster instances of the same verb according to their semantic meaning. Since all the targets of the clustering are the same verbs, there should be no difference in the results between the cases using v_{WORD} and v_{MASK} as embeddings. Therefore, we use only v_{MASK} for this process. We adopt X-means (Pelleg and Moore, 2000) or group average clustering based on a Euclidean distance as the clustering algorithm.

While X-means automatically determine the number of clusters, group average clustering requires a clustering termination threshold. In the group average clustering, the distance between two

clusters is defined as the average distances of all instance pairs between clusters, and the cluster pairs with the smallest distance between clusters are merged in order. The clustering is terminated when there are no more cluster pairs for which the distance between two clusters is less than or equal to a threshold θ. In this study, θ is shared across verbs, not determined for each verb. Note that when θ is set to a sufficiently large value, the number of clusters is one for all verbs. To set θ to an appropriate value, we gradually decrease θ from a sufficiently large value and fix it to a value where the number of the generated frame clusters is equal to the actual number of frames in the development set.

In the theory of Frame Semantics (Fillmore, 2006) on which FrameNet is based, the association between a word and a semantic frame is called a lexical unit (LU). Since each cluster generated as the result of clustering in the first step is a set of instances of the same verb used in the same meaning, it can be considered to correspond to an LU. Therefore, we refer to it as pseudo-LU (pLU).

Clustering across Verbs The clustering in the second step aims to cluster the pLUs generated as the result of the first-step clustering across verbs according to their meaning. This step calculates average contextualized embeddings of each pLU and then clusters the pLUs by using the calculated embeddings across verbs. We adopt Ward clustering or group average clustering based on a Euclidean distance as the clustering algorithm.

We need a termination criterion for both clustering algorithms. A straightforward approach is to use the ratio of the number of frames to the number of verbs. However, this approach does not work well in this case since there is an upper limit to the number of frame types and the number of frames to be generated does not increase linearly with the number of verbs. Therefore, in this study, we use the ratio of pLU pairs belonging to the same cluster as the termination criterion. Specifically, the clustering is terminated when the ratio of pLU pairs belonging to the same cluster $p_{\mathrm{F}_1=\mathrm{F}_2}$ is greater than or equal to the ratio of LU pairs belonging to the same frame in the development set $p_{\mathrm{C}_1=\mathrm{C}_2}$. Here, $p_{\mathrm{F}_1=\mathrm{F}_2}$ is calculated as:

$$p_{\mathrm{F}_1=\mathrm{F}_2} = \frac{\text{\# of pLU pairs in the same cluster}}{\text{\# of all pLU pairs}}. \quad (2)$$

While the number of all pLU pairs is constant regardless of clustering process, the number of pLU pairs belonging to the same cluster monotonically increases as the clustering process progresses. $p_{\mathrm{C}_1=\mathrm{C}_2}$ can be calculated as well as $p_{\mathrm{F}_1=\mathrm{F}_2}$ and $p_{\mathrm{C}_1=\mathrm{C}_2}$ reaches 1 when the number of the entire cluster becomes one cluster. Therefore, $p_{\mathrm{C}_1=\mathrm{C}_2}$ is guaranteed to be greater than or equal to $p_{\mathrm{F}_1=\mathrm{F}_2}$ during the clustering process. Since the probability that randomly selected LU pairs belong to the same frame is not affected by the data size, the criterion is considered valid regardless of the data size.

	#Verbs	#LUs	#Frames	#Examples
Dev.	255	300	169	12,718
Test	1,017	1,188	393	47,499
All	1,272	1,488	434	60,217

Table 2: Statistics of the dataset from FrameNet.

3 Experiment

We conducted an experiment of semantic frame induction to confirm the efficacy of our method. In this experiment, the objective is to group the given frame-evoking verbs with their context according to the frames they evoke.

3.1 Setting

Dataset From Berkeley FrameNet data release 1.7[1] in English, we extracted verbal LUs with at least 20 example sentences and used their example sentences. That is, all target verbs in the dataset have at least 20 example sentences for each frame they evoke. We limited the maximum number of sentence examples for each LU to 100 and if there were more examples, we randomly selected 100. Note that we did not use the SemEval-2019 Task 2 dataset because the dataset is no longer available as described on the official web page.[2]

The extracted dataset contained 1,272 different verbs as frame-evoking words. We used the examples for 255 verbs (20%) as the development set and those for the remaining 1,017 verbs (80%) as the test set. Thus, there are no overlapping frame-evoking verbs or LUs between the development and test sets, but there is an overlap in the frames evoked. We divided the development and test sets so that the proportion of verbs that evoke more than one frames would be the same. The development set was used to determine the alpha of $v_{\mathrm{W+M}}$

[1] https://framenet.icsi.berkeley.edu/

[2] https://competitions.codalab.org/competitions/19159#learn_the_details-datasets

Model	Clustering		α	#pLU	#C	PU / iPU / PiF	BcP / BcR / BcF
1-cluster-per-head	1cpv		–	–	1017	88.9 / 39.7 / 54.9	86.6 / 33.9 / 48.7
Arefyev et al. (2019)	GA (Cosine)		–	–	995	69.9 / 55.1 / 61.6	62.8 / 44.0 / 51.7
Anwar et al. (2019)	GA (Manhattan)		–	–	891	71.5 / 52.0 / 60.2	65.1 / 41.0 / 50.3
Ribeiro et al. (2019)	Chinese Whispers		–	–	542	50.9 / 66.3 / 57.5	39.4 / 56.7 / 46.5
One-step clustering	Ward		0.0	–	393	64.3 / 49.5 / 56.0	55.2 / 38.9 / 45.6
	GA		0.0	–	393	38.7 / 64.9 / 48.5	26.1 / 52.5 / 34.9
Two-step clustering	**first-step**	**second-step**					
	1cpv'	Ward	0.8	1017	164	54.8 / 73.1 / 62.7	43.1 / 64.3 / 51.6
	1cpv'	GA	0.9	1017	412	69.0 / 71.3 / 70.1	60.5 / 62.3 / 61.4
	GA	Ward	0.9	1196	291	49.3 / 72.9 / 58.8	37.3 / 64.6 / 47.3
	GA	GA	0.6	1196	479	63.0 / **76.3** / 69.0	52.8 / **68.0** / 59.4
	X-means	Ward	0.8	1043	167	54.0 / 72.2 / 61.8	42.6 / 63.6 / 51.1
	X-means	GA	0.7	1043	410	**71.9** / 74.1 / **73.0**	**63.2** / 65.5 / **64.4**

Table 3: Experimental results. #pLU denotes the number of pLUs and #C denotes the number of frame clusters. Note that the actual numbers of LUs and frames are 1,188 and 393, respectively. GA means group average clustering.

and the termination criterion for the clustering in each step and layers to be used as contextualized word embeddings. Table 2 lists the statistics of the dataset.

Models We compared four models, all combinations of group average clustering or X-means in the first step and Ward clustering or group average clustering in the second step. We also compared a model that treats all instances of one verb as one cluster (1-cluster-per-verb; 1cpv) and models that treat all instances of one verb as one cluster (1cpv') in the first step and then perform the clustering in the second step.

In addition, we compared our models with the top three models in Subtask-A of SemEval-2019 Task 2. Arefyev et al. (2019) first perform group average clustering using BERT embeddings of frame-evoking verbs. Then, they perform clustering to split each cluster into two by using TF-IDF features with paraphrased words. Anwar et al. (2019) use the concatenation of the embedding of a frame-evoking verb and the average word embedding of all words in a sentence obtained by skip-gram (Mikolov et al., 2013). They perform group average clustering based on Manhattan distance by using the embedding. Ribeiro et al. (2019) perform graph clustering based on Chinese whispers (Biemann, 2006) by using ELMo embeddings of frame-evoking verbs.

To confirm the usefulness of the two-step clustering, we also compared our models with models that perform a one-step clustering. For the model, we used Ward clustering or group average clustering as the clustering method and $v_{\text{W+M}}$ as the contextualized word embedding. We gave the oracle number of clusters to these models, i.e., we stopped clustering when the number of human-annotated frames and the number of cluster matched.

Metrics and Embeddings We used six evaluation metrics: B-CUBED PRECISION (BCP), B-CUBED RECALL (BCR), and their harmonic mean, F-SCORE (BCF) (Bagga and Baldwin, 1998), and PURITY (PU), INVERSE PURITY (IPU), and their harmonic mean, F-SCORE (PIF) (Karypis et al., 2000). We used BERT (bert-base-uncased) in Hugging Face[3] as the contextualized word embedding.

3.2 Results

Table 3 shows the experimental results.[4] When focusing on BCF, which was used to rank the systems in Subtask-A of SemEval-2019 Task 2, our model using X-means as the first step and group average clustering as the second step achieved the highest score of 64.4. It also got the highest PIF score of 73.0. The number of human-annotated frames was 393, while the number of generated clusters was 410. These results demonstrate that the termination criterion of the two-step clustering works effectively.

In all two-step clustering methods, α was tuned between 0.0 and 1.0, which shows that both v_{WORD} and v_{MASK} should be considered. In addition, α was close to 1.0 for these methods, which indicates that v_{MASK} is more useful for clustering instances across verbs. In contrast, $v_{\text{W+M}}$ in the one-step clustering methods was equivalent to v_{WORD} with $\alpha = 0.0$. This indicates that there is no effect of using v_{MASK}

[3] https://huggingface.co/transformers/

[4] The performance of the top three models in Subtask-A of SemEval-2019 Task 2 is lower than reported in the task because the dataset used in this study has a high proportion of verbs that evoke multiple frames and is, therefore, a challenging dataset.

for the one-step clustering-based methods.

The two-step clustering-based models that use group average clustering as the second clustering algorithm tended to achieve high scores. This indicates that the two-step clustering-based approach, which first cluster instances of the same verb and then cluster across verbs, is effective. However, as to the first clustering, 1cpv' strategy, which treats all the instances of the same verb as one cluster, achieved a higher accuracy than the clustering of the group average method, and achieved an accuracy close to the clustering of X-means, and thus we can say that 1cpv' strategy is effective enough for this dataset. We think this is due to the fact that the dataset used in this study is quite biased towards verbs that evoke only one frame, and we believe that the effectiveness of the 1cpv' may be limited in a more practical setting. Further investigation of this is one of our future works.

4 Conclusion

We proposed a method that uses masked word embeddings and two-step clustering for semantic frame induction. The results of experiments using FrameNet data showed that masked word embeddings and two-step clustering are quite effective for this frame induction task. We will conduct experiments in a setting where nouns and adjectives are also accounted for as frame-evoking words. The future goal of this research is to build a frame-semantic resource, which requires not only the induction of semantic frames but also the determination of the arguments required by each frame and the induction of semantic roles of the arguments. A possible extension of our approach is to utilize contextualized word embeddings of arguments of verbs to see if it is possible to generalize our approach for achieving this goal.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 18H03286 and 21K12012.

References

Saba Anwar, Dmitry Ustalov, Nikolay Arefyev, Simone Paolo Ponzetto, Chris Biemann, and Alexander Panchenko. 2019. HHMM at SemEval-2019 task 2: Unsupervised frame induction using contextualized word embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 125–129.

Nikolay Arefyev, Boris Sheludko, Adis Davletov, Dmitry Kharchev, Alex Nevidomsky, and Alexander Panchenko. 2019. Neural GRANNy at SemEval-2019 task 2: A combined approach for better modeling of semantic relationships in semantic frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 31–38.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING 1998)*, pages 79–85.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING 1998)*, pages 86–90.

Chris Biemann. 2006. Chinese whispers-an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing (TextGraphs 2006)*, pages 73–80.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 4171–4186.

Charles J Fillmore. 2006. Frame semantics. *Cognitive Linguistics: Basic Readings*, 34:373–400.

Michael Karypis, Steinbach George, and Vipin Kumar. 2000. A comparison of document clustering techniques. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS 2013)*, pages 3111–3119.

Dan Pelleg and Andrew Moore. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 727–734.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 2227–2237.

Behrang QasemiZadeh, Miriam R. L. Petruck, Regina Stodden, Laura Kallmeyer, and Marie Candito. 2019. SemEval-2019 task 2: Unsupervised lexical frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 16–30.

Eugénio Ribeiro, Vânia Mendonça, Ricardo Ribeiro, David Martins de Matos, Alberto Sardinha, Ana Lúcia Santos, and Luísa Coheur. 2019. L2F/INESC-ID at SemEval-2019 task 2: Unsupervised lexical semantic frame induction using contextualized word representations. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 130–136.

Lightweight Adapter Tuning for Multilingual Speech Translation

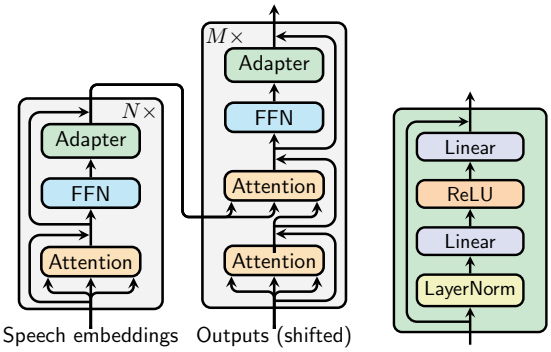
Hang Le[1] **Juan Pino**[2] **Changhan Wang**[2]
Jiatao Gu[2] **Didier Schwab**[1] **Laurent Besacier**[1,3]
[1]Univ. Grenoble Alpes, CNRS, LIG [2]Facebook AI [3]Naver Labs Europe
{hang.le, didier.schwab, laurent.besacier}@univ-grenoble-alpes.fr
{juancarabina, changhan, jgu}@fb.com

Abstract

Adapter modules were recently introduced as an efficient alternative to fine-tuning in NLP. Adapter tuning consists in freezing pre-trained parameters of a model and injecting lightweight modules between layers, resulting in the addition of only a small number of task-specific trainable parameters. While adapter tuning was investigated for multilingual neural machine translation, this paper proposes a comprehensive analysis of adapters for multilingual speech translation (ST). Starting from different pre-trained models (a multilingual ST trained on parallel data or a multilingual BART (mBART) trained on non-parallel multilingual data), we show that adapters can be used to: (a) efficiently specialize ST to specific language pairs with a low extra cost in terms of parameters, and (b) transfer from an automatic speech recognition (ASR) task and an mBART pre-trained model to a multilingual ST task. Experiments show that adapter tuning offer competitive results to full fine-tuning, while being much more parameter-efficient.

1 Introduction

The question of *versatility* versus *specialization* is often raised in the design of any multilingual translation system: is it possible to have a single model that can translate from any source language to any target one, or does it have to be multiple models each of which is in charge of one language pair? The former is referred to as a *multilingual* model, while the latter are *bilingual* ones. These two paradigms have their own strengths and limitations. From a practical point of view, a multilingual model seems to be highly desirable due to its simplicity in *training* and *deployment*, in terms of both time and space complexities. However, in terms of *accuracy*, a multilingual model could be outperformed by its bilingual counterparts, especially on high-resource language pairs.



(a) Transformer with adapters. (b) An adapter cell.

Figure 1: (a) Transformer with adapters at its FFN sub-layers. For simplicity, layer normalization (Ba et al., 2016) is omitted. During fine-tuning, only the adapters are trained. (b) A typical adapter architecture.

In practice, a certain trade-off between the aforementioned factors (and thus more generally between versatility and specialization) has often to be made, and depending on the application, one can be favored more than the other. One way to move along the spectrum between multilingual and bilingual models is to use adapter tuning which consists in freezing pre-trained parameters of a multilingual model and injecting lightweight modules between layers resulting in the addition of a small number of language-specific trainable parameters. While adapter tuning was investigated for multilingual neural machine translation (NMT) (Bapna and Firat, 2019), to our knowledge, this paper proposes the first comprehensive analysis of adapters for multilingual speech translation.

Our contributions are the following: (1) we show that both versatility and specialization can be achieved by tuning language-specific adapter modules on top of a multilingual system. Bilingual models with higher accuracy than the original multilingual model are obtained, yet keeping a low maintenance complexity; (2) starting from a different initialization point, we show that adapters can also be

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 817–824
August 1–6, 2021. ©2021 Association for Computational Linguistics

used as a glue to connect off-the-shelf systems (an automatic speech recognition (ASR) model and a multilingual denoising auto-encoder mBART (Liu et al., 2020; Tang et al., 2020)) to perform the multilingual ST task. Extensive experiments on the MuST-C dataset (Di Gangi et al., 2019) show that adapter-based fine-tuning can achieve very competitive results to full fine-tuning—while being much more parameter-efficient—in both standard and low-resource settings. Our code based on FAIRSEQ S2T (Wang et al., 2020) is publicly available.[1]

2 Related Work

Adapter layers (or *adapters* for short) were first proposed in computer vision (Rebuffi et al., 2017), then explored for text classification tasks in NLP (Houlsby et al., 2019). Adapters are generally inserted between the layers of a pre-trained network and finetuned on the adaptation corpus. Bapna and Firat (2019) studied adapters in the context of NMT and evaluated them on two tasks: domain adaptation and massively multilingual NMT. Philip et al. (2020) later introduced monolingual adapters for zero-shot NMT. Other research groups made contributions on the use of adapters in NLP (Pfeiffer et al., 2020b, 2021) and a framework built on top of HuggingFace Transformers library (Wolf et al., 2020) was also released to facilitate the downloading, sharing, and adapting state-of-the-art pre-trained models with adapter modules (Pfeiffer et al., 2020a). Also very relevant to our paper is the work of Stickland et al. (2021) where adapters are used to adapt pre-trained BART (Lewis et al., 2020) and mBART25 (multilingual BART pre-trained on 25 languages) (Liu et al., 2020) to machine translation.

As far as speech processing is concerned, adapters were mostly used in ASR (Kannan et al., 2019; Lee et al., 2020; Winata et al., 2020; Zhu et al., 2020). Recently, they have also been explored for ST as well but in a limited scope. Escolano et al. (2020) addressed a very specific setting (zero-shot ST), while Li et al. (2020) used only a single adapter after a Transformer encoder.

3 Adapters for Speech Translation

In this section, we describe the integration of adapters into a given backbone model for speech translation. As the Transformer (Vaswani et al., 2017) has become increasingly common in speech processing,[2] it will be used as our backbone. Our method, however, can be easily applied to any other architectures, e.g., dual-decoder Transformer (Le et al., 2020).

Adapter modules can be introduced into a Transformer in a *serial* or *parallel* fashion. Consider a layer represented by a function f that produces an output $\mathbf{y}$ from an input $\mathbf{x}$, i.e., $\mathbf{y} = f(\mathbf{x})$. This can be an entire encoder or decoder layer, or just one of their sub-layers (e.g., the self-attention or the final feed-forward network (FFN) component). Suppose that our adapter layer is represented by a function g. The new "adapted" output is then given by:

$$\mathbf{y}_{\text{serial}} = g(f(\mathbf{x})), \quad \mathbf{y}_{\text{parallel}} = f(\mathbf{x}) + g(\mathbf{x}).$$

Intuitively, a serial adapter modifies the output directly, while a parallel one performs the operations in parallel before merging its output to the layer. In Figure 1a, we show an example of serial adapters being integrated to the Transformer, or more precisely to its FFN sub-layers. A common adapter module (Bapna and Firat, 2019) is presented in Figure 1b. Here g is a small FFN with a residual connection. The first linear layer is typically a down projection to a bottleneck dimension, and the second one projects the output back to the initial dimension. Bottleneck allows us to limit the number of parameters. Other adapter architectures also exist, e.g., Stickland and Murray (2019) explored parallel adapters consisting of a multi-head attention (MHA) layer in a multi-task setup.

For multilingual ST, we adopt the following general recipe for adapter-based fine-tuning. Starting from a pre-trained backbone, an adapter is added for each language pair and then finetuned on the corresponding bilingual data (while the rest of the backbone is frozen). The pre-trained backbone plays a crucial role in this recipe. We explore two common scenarios to obtain this pre-trained model, namely *refinement* and *transfer learning*. We present them in details, together with extensive experimental results, in Section 5 and 6. In the next section, we present our experimental setup.

4 Experimental Setup

4.1 Dataset

MuST-C We evaluate our recipes on MuST-C (Di Gangi et al., 2019), a large-scale one-to-many

[1] https://github.com/formiel/fairseq/tree/master/examples/speech_to_text/docs/adapters.md

[2] For speech applications (Inaguma et al., 2020; Wang et al., 2020), the embedding layer of the encoder is often a small convolutional neural network (Fukushima and Miyake, 1982; LeCun et al., 1989).

	Dict	D	Adapter d	Adapter ENC	Adapter DEC	Finetune ENC	Finetune DEC	# params (M) trainable/total	de	es	fr	it	nl	pt	ro	ru	avg
								Training data (hours)	408	504	492	465	442	385	432	489	
1	mono	256	-	-	-	-	-	8×31.1/8×31.1	22.16	30.42	27.92	22.92	24.10	27.19	21.51	14.36	23.82
2	multi		-	-	-	-	-	32.1/32.1	22.37	30.40	27.49	22.79	24.42	27.32	20.78	14.54	23.76
3	multi		64	-	✓	-	-	8×0.2/33.7	22.32	30.50	27.55	22.91	24.51	27.36	21.09	14.74	23.87
4	multi		64	✓	✓	-	-	8×0.6/36.9	22.75	31.07	28.03	23.04	24.75	28.06	21.20	14.75	24.21
5	multi		128	-	✓	-	-	8×0.4/35.3	22.45	30.85	27.71	23.06	24.57	27.52	20.93	14.57	23.96
6	multi		128	✓	✓	-	-	8×1.2/41.7	22.84*	31.25*	28.29*	23.27*	24.98*	28.16*	21.36*	14.71	24.36
7	multi		-	-	-	-	✓	8×14.6/8×32.1	23.49	31.29	28.40	23.63	25.51	28.71	21.73	15.22	24.75
8	multi		-	-	-	✓	✓	8×32.1/8×32.1	23.13*	31.39*	28.67*	23.80*	25.52*	29.03*	22.25*	15.44*	24.90
9	mono	512	-	-	-	-	-	8×74.3/8×74.3	21.93	30.46	27.90	22.64	23.98	25.98	20.50	14.01	23.42
10	multi		-	-	-	-	-	76.3/76.3	23.98	32.47	29.24	24.97	26.20	29.81	22.74	15.30	25.59
11	multi		64	-	✓	-	-	8×0.4/79.5	24.24	32.52	29.47	24.74	26.13	29.72	22.53	15.25	25.57
12	multi		64	✓	✓	-	-	8×1.2/85.9	24.13	32.80	29.55	24.90	26.04	30.25	22.73	15.31	25.72
13	multi		128	-	✓	-	-	8×0.8/82.7	24.34	32.86	29.51	24.73	26.15	30.01	22.58	15.07	25.66
14	multi		128	✓	✓	-	-	8×2.4/95.5	24.30	32.61	29.72*	25.07	26.29	30.46*	22.99	15.47	25.86
15	multi		256	-	✓	-	-	8×1.6/89.1	24.38	32.78	29.69	24.72	26.25	29.93	22.63	15.40	25.72
16	multi		256	✓	✓	-	-	8×4.8/114.7	24.61	32.94	29.67	**25.12**	26.16	**30.53**	22.66	15.31	25.88
17	multi		-	-	-	-	✓	8×35.5/8×36.3	**24.67**	**33.12**	**30.11**	25.05	**26.33**	29.85	**23.04**	**15.61**	**25.97**
18	multi		-	-	-	✓	✓	8×76.3/8×76.3	24.54*	32.95*	29.96*	25.01	26.31	30.04	22.66	15.54*	25.88

Table 1: BLEU on MuST-C dev set for **refinement**. In the **Dict** column, mono and multi mean, respectively, monolingual and multilingual dictionary. D is the Transformer hidden dimension. In the **Adapter** group, d is the adapter bottleneck dimension, ENC and DEC mean adding adapters to encoder and decoder, respectively; and idem for the **Finetune** group. Rows 1–2 and rows 9–10 represent our bilingual and multilingual baselines for each D. Values lower than the multilingual baselines are colored in blue. The highest values in each group of D are underlined, while the highest values of each column are in **bold** face. Furthermore, we select the top configurations (6, 8, 14, 18) and perform statistical significance test using bootstrap re-sampling (Koehn, 2004). Results passing the test (compared to the corresponding multilingual baselines, with p-value < 0.05) are marked with a star.

ST dataset from English to eight target languages including Dutch (nl), French (fr), German (de), Italian (it), Portuguese (pt), Romanian (ro), Russian (ru), and Spanish (es). Each direction includes a triplet of speech, transcription, and translation. Sizes range from 385 hours (pt) to 504 hours (es).

MuST-C-Imbalanced We built a low-resource version of MuST-C, called MuST-C-Imbalanced, in which we randomly keep only $X\%$ of the original training data, where $X = 100$ for es, fr; $X = 50$ for ru, it; $X = 20$ for nl, ro; and $X = 10$ for de, pt (same order of the languages in the original MuST-C if we sort them in decreasing amount of data). The amount of speech data ranges from 41 hours (de) to 504 hours (es) in this version, better reflecting real-world data imbalance scenarios.

4.2 Implementation details

Our implementation is based on the FAIRSEQ S2T toolkit (Wang et al., 2020). We experiment with two architectures: a small Transformer model with dimension $D = 256$ and a medium one where $D = 512$. All experiments use the same encoder with 12 layers. The decoder has 6 layers, except for the transfer learning scenario where we used the mBART decoder for initialization. We used 8k and 10k unigram vocabulary (Kudo and Richardson, 2018) for bilingual and multilingual models, respectively. The speech features are 80-dimensional log mel filter-bank. Utterances having more than 3000 frames are removed for GPU efficiency. We used SpecAugment (Park et al., 2019) with LibriSpeech basic (LB) policy for data augmentation.

We used the Adam optimizer (Kingma and Ba, 2015) with learning rate linearly increased for the first 10K steps to a value $\eta_{\max}$, then decreased proportionally to the inverse square root of the step counter. For all adapter experiments, $\eta_{\max}$ is set to $2\text{e}{-3}$. For the others, however, we perform a grid search over three values $\{2\text{e}{-3}, 2\text{e}{-4}, 2\text{e}{-5}\}$ and select the best one on the dev set, as they are more sensitive to the learning rate.

5 Refinement

In this section, a fully trained multilingual ST backbone is further refined on each language pair to boost the performance and close potential gaps with bilingual models. We compare adapter tuning with other fine-tuning approaches as well as the bilingual and multilingual baselines (the latter being the starting point for all fine-tuning approaches) (Bapna and Firat, 2019). Starting from

	D	Adapter d	Adapter ENC	Adapter DEC	Finetune ENC	Finetune DEC	# params (M) trainable/total	de	es	fr	it	nl	pt	ro	ru	avg
							Training data (hours)	41	504	492	232	89	38	86	245	
1	256	-	-	-	-	-	32.1/32.1	15.99	30.51	28.17	21.80	20.27	22.47	17.38	13.18	21.22
2		128	✓	✓	-	-	8×1.2/41.7	17.02	30.71	28.42	22.37	21.01	23.74	18.55	13.52	21.92
3		-	-	-	✓	✓	8×32.1/8×32.1	16.93	30.86	28.34	22.42	20.86	23.44	18.49	13.63	21.87
4	512	-	-	-	-	-	76.3/76.3	17.05	31.92	29.06	22.91	21.64	24.15	19.18	14.09	22.50
5		256	✓	✓	-	-	8×4.8/114.7	17.46	**31.94**	29.09	**23.11**	21.76	**24.96**	**19.50**	14.10	**22.74**
6		-	-	-	✓	✓	8×76.3/8×76.3	**17.49**	31.67	**29.27**	22.97	**21.80**	24.80	19.43	**14.17**	22.70

Table 2: BLEU on MuST-C dev set for **refinement** in the low-resource scenario where the models were trained on MuST-C-Imbalanced dataset. We refer to Table 1 for other notation.

	Method	# params (M) trainable/total	de	es	fr	it	nl	pt	ro	ru	avg
Ours	Baseline	76.3/76.3	24.18	28.28	**34.98**	24.62	**28.80**	**31.13**	23.22	15.88	26.39
	Best adapting	8 × 4.8/76.3	**24.63**	**28.73**	34.75	**24.96**	**28.80**	30.96	23.70	**16.36**	**26.61**
	Best fine-tuning	8 ×35.5/8 × 76.3	24.50	28.67	34.89	24.82	28.38	30.73	**23.78**	16.23	26.50
Li et al.	LNA-D	53.5/76.3	24.16	28.30	34.52	24.46	28.35	30.51	23.29	15.84	26.18
	LNA-E	48.1/76.3	24.34	28.25	34.42	24.24	28.46	30.53	23.32	15.89	26.18
	LNA-E,D	25.3/76.3	24.27	28.40	34.61	24.44	28.25	30.53	23.27	15.92	26.21

Table 3: BLEU on MuST-C test set. Our method compares favorably with (Li et al., 2020).

these backbones, we either add language-specific adapters and train them only, or we finetune the backbone on each language pair, either fully or partially. All these trainings are performed on MuST-C. The results are shown in Table 1. There are two main blocks corresponding to two architectures: $D = 256$ (small) and $D = 512$ (medium). Rows 1 and 9 provide the bilingual baselines, while rows 2 and 10 serve as the multilingual baselines for each block. In addition, we compare adapter-tuning with full fine-tuning and multilingual-training (baseline) on MuST-C-Imbalanced. Table 2 displays the results for this set of experiments.

Bilingual vs. Multilingual For the small architecture ($D = 256$), the bilingual models slightly outperform their multilingual counterpart (rows 1, 2). Looking further into the performance of each language pair, the multilingual model is able to improve the results for 4 out of 8 pairs (de, nl, pt, ru), mainly those in the lower-resource direction, but the joint multilingual training slightly hurts the performance of higher-resource pairs such as es, fr, it, and ro. Finally, we observe that the medium model ($D = 512$) performs better in the multilingual setting than the bilingual one (rows 9, 10).

Adapter tuning vs. Fine-tuning Both recipes yield improvements over the multilingual baseline and recover the lost performance of higher-resource directions compared to the bilingual baseline for the small model ($D = 256$). For the medium one ($D = 512$), one adapter tuning (row 14) can slightly improve the scores in all directions and even approach the results of the best fine-tuning experiment (row 17) while maintaining much lower model sizes (95.5M vs. 8× 36.3M parameters).

Low-resource scenario The obtained results on small models show that adapter-tuning achieved the best performance, producing clear improvements over the baseline, especially for the low-resource languages: +1.1 BLEU on average on nl, ro, de, pt; +0.3 BLEU on average on es, fr, ru, it; which is competitive to full fine-tuning (+0.9 and +0.4 BLEU, respectively) while being more parameter-efficient as well as simpler for training and deployment (one model with adapters versus eight separate models). For larger models, however, the improvement is smaller: +0.4 BLEU on average on the lower-resource pairs and +0.1 on the higher-resource ones; while those of full fine-tuning are +0.4 and roughly no improvement, respectively.

Results on test set We select the best-performing fine-tuning recipes on the dev set (rows 16 and 17 in Table 1) for evaluation on the test set. For reference, we also include the multilingual baseline (row 10). Moreover, to go beyond conventional fine-tuning approaches, we also compare our recipes with a contemporary work in which only several components of the network are finetuned (Li et al., 2020). For a fair comparison, we did not use large pre-trained components such as wav2vec (Baevski et al., 2020) or mBART (Tang et al., 2020) but instead considered the same pre-trained compo-

	Adapter			Finetune	# params (M)								
	d	ENC	DEC	xattn	trainable/total	de	es	fr	it	nl	pt	ro	ru	avg
1	-	-	-	-	8×31.1/8×31.1	22.16	30.42	27.92	22.92	24.10	27.19	21.51	14.36	23.82
2	-	-	-	✓	38 / 486	18.41	25.42	23.46	18.44	20.87	20.55	17.19	11.79	19.52
3	512	-	✓	-	101 / 587	0.94	0.65	0.93	0.76	0.95	0.89	0.52	0.93	0.82
4	512	-	✓	✓	139 / 587	21.98	29.47	27.05	22.89	24.06	26.34	21.0	14.35	23.39
5	512	✓	✓	-	152 / 638	11.04	18.62	16.10	12.37	13.18	14.29	10.62	6.95	12.90
6	512	✓	✓	✓	190 / 638	22.62	30.85	28.23	23.09	24.43	26.56	22.13	14.92	24.10

Table 4: BLEU on MuST-C dev set for **transfer learning** from pre-trained ASR and mBART models. We compare the results with the bilingual baselines (trained from scratch), shown in row 1 (which is identical to row 1 in Table 1). The column "Finetune xattn" means updating the cross-attention parameters. We refer to Table 1 for other notation.

nents used in our previous experiments. Following (Li et al., 2020), we considered six variants: fine-tuning LayerNorm + Attention in the encoder (LNA-E), or the decoder (LNA-D), or both (LNA-E,D); each with or without the length adapter. We found that adding the length adapter did not help in our experiments. Table 3 shows that our approach compares favorably with (Li et al., 2020) in terms of both performance and parameter-efficiency.

Other comments For small models, the encoder adapters boost the performance (0.3–0.4 BLEU on average) in all directions (rows 3 and 4, 5 and 6, Table 1), indicating that language-specific adapters can tweak the encoder representations to make them better suited for the decoder. In larger models, however, the impact of the encoder adapters is varied depending on languages and bottleneck dimensions. We also notice that increasing the bottleneck dimension slightly improves performance while remaining parameter-efficient. Fine-tuning remains the best option to optimize the models in most cases but leads to much larger model sizes. The adapter-tuning approach is competitive to fine-tuning while being much more parameter-efficient.

6 Transfer Learning

In this section, we show that adapters can be used to combine available pre-trained models to perform a multilingual ST task. In particular, we initialize the encoder using a pre-trained ASR encoder (on MuST-C)[3] provided by Wang et al. (2020) and the decoder using mBART50, a multilingual denoising auto-encoder pre-trained on 50 languages (Tang et al., 2020). We tune language independent cross-attention and language-specific adapters on top of these backbone models (using MuST-C as well). The results presented in Table 4 highlight that fine-tuning cross-attention is crucial to transfer to multilingual ST (rows 3 and 5 show poor results without doing so). Adding adapters to the backbone decoder (row 4) or to both encoder and decoder (row 6) further boosts performance, demonstrating the ability of adapters to connect off-the-shelf models in a modular fashion. The best-performing model in this recipe (row 6) also outperforms bilingual systems (row 1) despite having fewer trainable parameters (190M vs. 248M). It is also important to mention that while we experiment on 8 target languages of MuST-C corpus, the multilingual ST model of row 2 should be practically able to decode into 50 different target languages. Investigating such a zero-shot ST scenario is left for future work.

[3]Pre-training on ASR data and then transferring to ST is not new but rather standard. See, e.g., Bansal et al. (2019).

7 Conclusion

We have presented a study of adapters for multilingual ST and shown that language-specific adapters can enable a fully trained multilingual ST model to be further specialized in each language pair. With these adapter modules, one can efficiently obtain a single multilingual ST system that outperforms the original multilingual model as well as multiple bilingual systems while maintaining a low storage cost and simplicity in deployment. In addition, adapter modules can also be used to connect available pre-trained models such as an ASR model and a multilingual denoising auto-encoder to perform the multilingual speech-to-text translation task.

Acknowledgments

This work was supported by a Facebook AI SRA grant, and was granted access to the HPC resources of IDRIS under the allocation 2020-AD011011695 made by GENCI. It was also done as part of the Multidisciplinary Institute in Artificial Intelligence MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). We thank the anonymous reviewers for their insightful questions and feedback.

References

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *NAACL-HLT (1)*, pages 58–68. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *EMNLP/IJCNLP (1)*, pages 1538–1548. Association for Computational Linguistics.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2012–2017. Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Carlos Segura. 2020. Enabling zero-shot multilingual spoken language translation with language-specific encoders and decoders. *CoRR*, abs/2011.01097.

Kunihiko Fukushima and Sei Miyake. 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Enrique Yalta Soplin, Tomoki Hayashi, and Shinji Watanabe. 2020. Espnet-st: All-in-one speech translation toolkit. *arXiv preprint arXiv:2004.10234*.

Anjuli Kannan, Arindrima Datta, Tara N. Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee. 2019. Large-scale multilingual speech recognition with a streaming end-to-end model. In *INTERSPEECH*, pages 2130–2134. ISCA.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*. Association for Computational Linguistics.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Taewoo Lee, Min-Joong Lee, Tae Gyoon Kang, Seokyeoung Jung, Minseok Kwon, Yeona Hong, Jungin Lee, Kyoung-Gu Woo, Ho-Gyeong Kim, Jiseung Jeong, et al. 2020. Adaptable multi-domain language model for transformer asr. *arXiv preprint arXiv:2008.06208*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2020. Multilingual speech translation with efficient finetuning of pretrained models. *arXiv e-prints*, pages arXiv–2010.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2613–2617. ISCA.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. In *EACL*, pages 487–503. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *EMNLP (Demos)*, pages 46–54. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In *EMNLP (1)*, pages 7654–7673. Association for Computational Linguistics.

Jerin Philip, Alexandre Bérard, Laurent Besacier, and Matthias Gallé. 2020. Language adapters for zero-shot neural machine translation. In *EMNLP 2020*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2021. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. In *EACL*, pages 3440–3453. Association for Computational Linguistics.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *ICML*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: fast speech-to-text modeling with fairseq. In *AACL/IJCNLP (System Demonstrations)*, pages 33–39. Association for Computational Linguistics.

Genta Indra Winata, Guangsen Wang, Caiming Xiong, and Steven Hoi. 2020. Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition. *arXiv preprint arXiv:2012.01687*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (Demos)*, pages 38–45. Association for Computational Linguistics.

Yun Zhu, Parisa Haghani, Anshuman Tripathi, Bhuvana Ramabhadran, Brian Farris, Hainan Xu, Han Lu, Hasim Sak, Isabel Leal, Neeraj Gaur, et al. 2020. Multilingual speech recognition with self-attention structured parameterization. *Proc. Interspeech 2020*, pages 4741–4745.

A Parallel Adapters

In this section, we present our preliminary experiments in which we explore different positions of the parallel adapters: in parallel with either Transformer layers or their sub-layers. We perform experiments where the adapters are added to the decoder. The results are shown in Table 5.

	Adapter			# params (M)	
	d	h	type	trainable/total	en-de
1	-	-	-	32.1/32.1	22.37
2	128	-	`ser`	0.4/32.5	22.45
3	128	4	`par-TL`	0.8/32.9	21.67
4	128	4	`par-SA`	0.8/32.9	19.55
5	128	4	`par-XA`	0.8/32.9	19.22

Table 5: BLEU on dev set for **parallel vs. serial adapters**. In the "Adapter" block, d is the adapter's dimension, h is the number of heads, `ser` stands for serial adapters, and `par` stands for parallel ones. The suffixes denote the position of the parallel adapters: in parallel with the Transformer layer (`TL`), or with self-attention sub-layer (`SA`), or with cross-attention sub-layer (`XA`).

Among the parallel variants, the one that performs operations in parallel with a full layer produces the best result. However, its performance still could not surpass the serial adapter (row 2) as well as the starting point (row 1).

B Specializing

In addition to the refinement recipe where language-specific adapters tailor the frozen multilingual ST model to translate in the corresponding direction, we also propose a recipe to facilitate the specialization in individual language pairs: by replacing the multilingual vocabulary by the monolingual ones corresponding to each target language. This recipe allows us to transfer from multilingual models to monolingual ones. A practical benefit is that one can easily leverage pre-trained multilingual models for new languages.

	Dict	D	Adapter d	Adapter ENC	Adapter DEC	Finetune ENC	Finetune DEC	# params (M) trainable/total	de	es	fr	it	nl	pt	ro	ru	avg
1	mono	256	-	-	-	-	-	8×31.1/8×31.1	22.16	30.42	27.92	22.92	24.10	27.19	21.51	14.36	23.82
2	multi		-	-	-	-	-	32.1/32.1	22.37	30.40	27.49	22.79	24.42	27.32	20.78	14.54	23.76
3	mono		64	-	✓	-	-	8×4.3/8×31.3	23.28	30.95	28.31	23.25	24.76	27.84	21.55	14.60	24.32
4	mono		64	✓	✓	-	-	8×4.7/8×31.7	23.53	31.16	28.83	23.29	24.43	28.18	21.38	14.66	24.44
5	mono		128	-	✓	-	-	8×4.5/8×31.5	23.33	31.05	28.67	23.43	24.83	28.10	21.44	14.58	24.43
6	mono		128	✓	✓	-	-	8×5.3/8×32.3	22.09	30.09	27.63	22.53	24.24	27.09	20.36	14.19	23.53
7	mono		-	-	-	-	✓	8×13.6/8×31.1	24.03	31.79	29.64	24.16	25.55	28.92	22.11	15.00	25.15
8	mono		-	-	-	✓	✓	8×31.1/8×31.1	23.89	31.72	29.23	23.65	25.14	28.23	21.83	14.80	24.81
9	mono	512	-	-	-	-	-	8×74.3/8×74.3	21.93	30.46	27.90	22.64	23.98	25.98	20.5	14.01	23.42
10	multi		-	-	-	-	-	76.3/76.3	23.98	32.47	29.24	24.97	26.20	29.81	22.74	15.30	25.59
11	mono		64	-	✓	-	-	8×8.6/8×74.7	23.85	31.79	29.63	24.26	25.77	28.97	22.18	15.02	25.18
12	mono		64	✓	✓	-	-	8×9.4/8×75.5	23.74	31.62	29.44	24.02	25.56	29.23	22.25	15.39	25.16
13	mono		128	-	✓	-	-	8×9.0/8×75.1	23.91	32.05	29.47	24.08	25.86	29.28	22.30	15.28	25.28
14	mono		128	✓	✓	-	-	8×10.6/8×76.7	23.98	32.28	29.40	24.46	25.46	29.28	21.90	15.15	25.24
15	mono		256	-	✓	-	-	8×9.8/8×75.9	23.91	32.12	29.45	24.17	25.67	29.01	22.31	15.37	25.25
16	mono		256	✓	✓	-	-	8×13/8×79.1	24.39	32.33	29.46	24.07	25.72	29.84	22.07	15.25	25.39
17	mono		-	-	-	-	✓	8×33.4/8×74.3	24.95	32.85	30.33	25.02	26.08	29.97	23.01	15.69	25.99
18	mono		-	-	-	✓	✓	8×74.3/8×74.3	24.77	32.35	30.14	24.79	25.79	29.85	22.71	15.77	25.77

Table 6: BLEU on MuST-C dev set for **specialization**. We refer to Table 1 for all notation.

Table 6 displays the results of the specializing recipe. Starting from a trained multilingual ST model, one can obtain an improvement of 1.3–1.4 BLEU on average (row 8 vs. row 1 and 2) compared to the bilingual and multilingual baselines trained from scratch for the small architecture where $D = 256$. However, for a larger network ($D = 512$), the gain is more modest (0.4 BLEU on average).

Parameter Selection: Why We Should Pay More Attention to It

Jie-Jyun Liu[1], Tsung-Han Yang[1], Si-An Chen[1,2], and Chih-Jen Lin[2]

[1]ASUS Intelligent Cloud Services
[2]National Taiwan University
{eleven1_liu,henry1_yang}@asus.com
{d09922007,cjlin}@csie.ntu.edu.tw

Abstract

The importance of parameter selection in supervised learning is well known. However, due to the many parameter combinations, an incomplete or an insufficient procedure is often applied. This situation may cause misleading or confusing conclusions. In this opinion paper, through an intriguing example we point out that the seriousness goes beyond what is generally recognized. In the topic of multi-label classification for medical code prediction, one influential paper conducted a proper parameter selection on a set, but when moving to a subset of frequently occurring labels, the authors used the same parameters without a separate tuning. The set of frequent labels became a popular benchmark in subsequent studies, which kept pushing the state of the art. However, we discovered that most of the results in these studies cannot surpass the approach in the original paper if a parameter tuning had been conducted at the time. Thus it is unclear how much progress the subsequent developments have actually brought. The lesson clearly indicates that without enough attention on parameter selection, the research progress in our field can be uncertain or even illusive.

1 Introduction

The importance of parameter selection in supervised learning is well known. While parameter tuning has been a common practice in machine learning and natural language processing applications, the process remains challenging due to the huge number of parameter combinations. The recent trend of applying complicated neural networks makes the situation more acute. In many situations, an incomplete or an insufficient procedure for parameter selection is applied, so misleading or confusing conclusions sometimes occur. In this opinion paper, we present a very intriguing example showing that, without enough attention on parameter selection, the research progress in our field can be uncertain or even illusive.

In the topic of multi-label classification for medical code prediction, Mullenbach et al. (2018) is an early work applying deep learning. The evaluation was conducted on MIMIC-III and MIMIC-II (Johnson et al., 2016), which may be the most widely used open medical records. For MIMIC-III, besides using all 8,922 labels, they follow Shi et al. (2017) to check the 50 most frequently occurring labels. We refer to these two sets respectively as

MIMIC-III-full and MIMIC-III-50.

We will specifically investigate MIMIC-III-50. Based on Mullenbach et al. (2018), many subsequent works made improvements to push the state of the art. Examples include (Wang et al., 2018; Sadoughi et al., 2018; Xie et al., 2019; Tsai et al., 2019; Cao et al., 2020a,b; Ji et al., 2020; Teng et al., 2020; Chen, 2020; Vu et al., 2020; Dong et al., 2021).

For the data set MIMIC-III-full, Mullenbach et al. (2018) tuned parameters to find the model that achieves the best validation performance. However, when moving to check the set MIMIC-III-50, they *applied the same parameters without a separate tuning.* We will show that this decision had a profound effect. Many works directly copied values from Mullenbach et al. (2018) for comparison and presented superior results. However, as demonstrated in this paper, if parameters for MIMIC-III-50 had been separately tuned, the approach in Mullenbach et al. (2018) easily surpasses most subsequent developments. The results fully indicate that parameter selection is more important than what is generally recognized.

This paper is organized as follows. In Section 2, we analyze past results. The main investigation is in Section 3, while Section 4 provides some discussion. Some implementation details are in the appendix. Code and supplementary materials can be found at http://www.csie.ntu.edu.tw/~cjlin/papers/parameter_selection.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 825–830
August 1–6, 2021. ©2021 Association for Computational Linguistics

Table 1: Experimental results from Mullenbach et al. (2018). Macro-F1 and Micro-F1 are Macro-averaged and Micro-averaged F1 values, respectively. P@n is the precision at n, the fraction of the n highest-scored labels that are truly associated with the test instance.

(a) MIMIC-III-full: 8,922 labels

	Macro-F1	Micro-F1	P@8
CNN	0.042	0.419	0.581
CAML	0.088	0.539	0.709

(b) MIMIC-III-50: 50 labels.

	Macro-F1	Micro-F1	P@5
CNN	0.576	0.625	0.620
CAML	0.532	0.614	0.609

2 Analysis of Works that Compared with Mullenbach et al. (2018)

The task considered in Mullenbach et al. (2018) is to predict the associated ICD (International Classification of Diseases) codes of each medical document. Here an ICD code is referred to as a label. The neural network considered is

$$\begin{aligned} &\text{document} \rightarrow \text{word embeddings} \\ &\rightarrow\text{convolution} \rightarrow \text{attention} \rightarrow \text{linear layer}, \end{aligned} \tag{1}$$

where the convolutional operation was based on Kim (2014). A focus in Mullenbach et al. (2018) was on the use of attention, so they detailedly compared the two settings[1]

CNN: (1) without attention,
CAML: (1).

For the data set MIMIC-III-full, CAML, which includes an attention layer, was shown to be significantly better than CNN on all criteria; see Table 1a. However, for MIMIC-III-50, the subset of the 50 most frequent labels, the authors reported in Table 1b that CAML is not better than CNN.

The paper (Mullenbach et al., 2018) has been highly influential. By exactly using their training, validation, and test sets for experiments, many subsequent studies have proposed new and better approaches; see references listed in Section 1. Most of them *copied* the CNN and CAML results from (Mullenbach et al., 2018) as the baseline for comparison. Table 2 summarizes their superior results on MIMIC-III-50.[2]

While using the same MIMIC-III-50 set, these subsequent studies differ from Mullenbach et al. (2018) in various ways. They proposed sophisticated networks and may incorporate additional information (e.g., label description, knowledge graph of words, etc.). Further, they may change settings not considered as parameters for tuning in Mullenbach et al. (2018). For example, Mullenbach et al. (2018) truncated each document to have at most 2,500 tokens, but Vu et al. (2020) used 4,000.

3 Investigation

We investigate the performance of the CNN and CAML approaches in Mullenbach et al. (2018) for the set MIMIC-III-50. Some implementation details are left in supplementary materials.

3.1 Parameter Selection in Mullenbach et al. (2018)

Mullenbach et al. (2018) conducted parameter tuning on a validation set of MIMIC-III-full. By considering parameter ranges shown in Table 3, they applied Bayesian optimization (Snoek et al., 2012) to choose parameters achieving the highest precision@8 on the validation set; see the selected values in Table 3 and the definition of precision in Table 1. However, the following settings are fixed instead of being treated as parameters for tuning.

- Each document is truncated to have at most 2,500 tokens. Word embeddings are from the CBOW method (Mikolov et al., 2013) with the embedding size 100.
- The stochastic gradient method Adam implemented in PyTorch is used with its default setting. However, the batch size is fixed to be 16 and the learning rate is considered as a parameter. Binary cross-entropy loss is considered.
- The Adam method is terminated if the precision@8 does not improve for 10 epochs. The model achieving the highest validation preision@8 is used to predict the test set for obtaining results in Table 1a.

Interestingly, for the 50-label subset of MIMIC-III, Mullenbach et al. (2018) did not conduct a parameter-selection procedure. Instead, a decision was to *use the same parameters selected for the*

[1] After convolution, each word is still associated with a short vector and attention is a way to obtain a single vector for the whole document. For CNN where attention is not used, Mullenbach et al. (2018) followed Kim (2014) to select the maximal value across all words.

[2] We exclude papers that used the same MIMIC-III-50 set but did not list values in Mullenbach et al. (2018) for comparison. Anyway, results in these papers are not better than what we obtained in Section 3.

Table 2: MIMIC-III-50 results from past works that have *directly listed values* in Mullenbach et al. (2018) for comparison. For Macro-F1, please see a note on its definition in the appendix.

	Macro-F1	Micro-F1	P@5	Code	Notes
Baseline considered					
CNN (Mullenbach et al., 2018)	0.576	0.625	0.620	Y	
CAML (Mullenbach et al., 2018)	0.532	0.614	0.609	Y	
New network architectures					
MVC-LDA (Sadoughi et al., 2018)	0.597	0.668	0.644	N	multi-view convolutional layers
DACNM (Cao et al., 2020b)	0.579	0.641	0.616	N	dilated convolution
BERT-Large (Chen, 2020)	0.531	0.605	-	N	BERT model
MultiResCNN (Li and Yu, 2020)	0.606	0.670	0.641	Y	multi-filter convolution and residual convolution
DCAN (Ji et al., 2020)	0.615	0.671	0.642	Y	dilated convolution, residual connections
G-Coder without additional information (Teng et al., 2020)	-	0.670	0.637	N	multiple convolutional layers
LAAT (Vu et al., 2020)	0.666	0.715	0.675	Y	LSTM before attention
New network architectures + additional information (e.g., label description, label co-occurrence, label embeddings, knowledge graph, adversarial learning, etc.)					
LEAM (Wang et al., 2018)	0.540	0.619	0.612	Y	label embeddings used
MVC-RLDA (Sadoughi et al., 2018)	0.615	0.674	0.641	N	label description used
MSATT-KG (Xie et al., 2019)	0.638	0.684	0.644	N	knowledge graph
HyperCore (Cao et al., 2020a)	0.609	0.663	0.632	N	label co-occurrence and hierarchy used
G-Coder with additional information (Teng et al., 2020)	-	0.692	0.653	N	knowledge graph, adversarial learning
Results of our investigation in Section 3 are listed below for comparison (values averaged from Table 4)					
CNN	0.606	0.659	0.634	Y	parameter selection applied
CAML	0.635	0.684	0.651	Y	

Table 3: Parameter ranges considered in Mullenbach et al. (2018) and the values used.

Parameter	Range	Values used	
		CNN	CAML
d_c: # filters	50-500	500	50
k: filter size	2-10	4	10
q: dropout prob.	0.2-0.8	0.2	0.2
η: learning rate	0.0003, 0.0001, 0.003, 0.001	0.003	0.0001

full-label set. Further they switch to present precision@5 instead of precision@8 because on average each instance is now associated with fewer labels.

The decision of not separately tuning parameters for MIMIC-III-50, as we will see, has a profound effect. In fact, because in Table 1b CAML is slightly worse than CNN, Mullenbach et al. (2018) have suspected that a parameter tuning may be needed. They stated that "we hypothesize that this[3] is because the relatively large value of $k = 10$ for CAML leads to a larger network that is more suited to larger datasets; tuning CAML's hyperparameters on this dataset would be expected to improve performance on all metrics." However, it seems no subsequent works tried to tune parameters of CNN or CAML on MIMIC-III-50.

3.2 Reproducing Results in Mullenbach et al.

To ensure the correctness of our implementation, first we reproduce the results in Mullenbach et al. (2018) by considering the following two programs.

- The public code by Mullenbach et al. (2018) at `github.com/jamesmullenbach/caml-mimic`.
- Our implementation of CNN/CAML by following the description in Mullenbach et al. (2018). The code is part of our development

[3]Here "this" means that CAML is not better than CNN.

Table 4: MIMIC-III-50 results after parameter selection. We consider three random seeds, where 1,337 was used in Mullenbach et al. (2018). Under each seed, we select the five models achieving the best validation precision@5, use them to predict the test set, and report mean/variance.

	Seed	Macro-F1	Micro-F1	P@5
CNN	1337	0.608 ± 0.006	0.659 ± 0.005	0.634 ± 0.002
	1331	0.601 ± 0.013	0.660 ± 0.007	0.634 ± 0.003
	42	0.608 ± 0.007	0.658 ± 0.006	0.633 ± 0.003
CAML	1337	0.640 ± 0.004	0.686 ± 0.004	0.650 ± 0.002
	1331	0.631 ± 0.004	0.682 ± 0.003	0.651 ± 0.001
	42	0.634 ± 0.009	0.684 ± 0.004	0.651 ± 0.002

on a general multi-label text classification package LibMultiLabel.[4]

Parameters and the random seed used in Mullenbach et al. (2018) are considered; see Table 3.

After some tweaks, on one GPU machine both programs give *exactly the same results* in the following table

	Macro-F1	Micro-F1	P@5
CNN	0.585	0.626	0.617
CAML	0.532	0.610	0.609

Values are very close to those in Table 1b. The small difference might be due to that our GPUs or PyTorch versions are not the same as theirs.

We conclude that results in Mullenbach et al. (2018) are reproducible.

3.3 Parameter Selection for MIMIC-III-50

We apply the parameter-selection procedure in Mullenbach et al. (2018) for MIMIC-III-full to MIMIC-III-50; see details in Section 3.1. A difference is that, because training MIMIC-III-50 is faster than MIMIC-III-full, instead of using Bayesian optimization, we directly check a grid of parameters that are roughly within the ranges given in Table 3. Specifically, we consider

$$d_c = 50, 150, 250, 350, 450, 550$$
$$k = 2, 4, 6, 8, 10$$
$$q = 0.2, 0.4, 0.6, 0.8$$

Because Mullenbach et al. (2018) switched to report test precision@5 for MIMIC-III-50, for validation we also use precision@5.

To see the effect of random seeds, besides the one used in Mullenbach et al. (2018), we checked two other seeds 1,331 and 42, selected solely because they are the lucky numbers of an author.

[4] https://github.com/ASUS-AICS/LibMultiLabel

3.4 Results and Analysis

Table 4 shows CNN/CAML results after parameter selection and we have the following observations.

- Both CNN and CAML achieve better results than those reported in Table 1b by Mullenbach et al. (2018). The improvement of CAML is so significant that it becomes better than CNN.
- From details in supplementary materials, for some parameters (e.g., d_c and q for CAML), the selected values are very different from those used by Mullenbach et al. (2018). Thus parameters selected for MIMIC-III-full are not transferable to MIMIC-III-50 and a separate tuning is essential.
- Results are not sensitive to the random seeds.[5]
- A comparison with Table 2 shows that most subsequent developments cannot surpass our CAML results. Some are even inferior to CNN, which is the baseline of all these studies.
- We checked if subsequent developments conducted parameter selection. A summary is in the supplementary materials.

Based on our results, how much progress past works have made is therefore unclear.

4 Discussion and Conclusions

The intention of this paper is to provide constructive critiques of past works rather than place blame on their authors. For the many parameter combinations, it is extremely difficult to check them. However, what our investigation showed is that if resources or time are available, more attention should be paid to the parameter selection. For Mullenbach et al. (2018), as they have done a comprehensive selection on a super-set MIMIC-III-full, the same procedure on the simpler MIMIC-III-50 is

[5] CNN is slight more sensitive to seeds than CAML. More investigation is needed.

entirely feasible. The decision of not doing so leads to a weak baseline in the subsequent developments.

In conclusion, besides proposing new techniques such as sophisticated networks, more attention should be placed on the parameter selection. In the future this helps to ensure that strong baselines are utilized to check the progress.

References

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020a. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3105–3114.

Pengfei Cao, Chenwei Yan, Xiangling Fu, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020b. Clinical-coder: Assigning interpretable ICD-10 codes to Chinese clinical notes. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 294–301.

Yiyun Chen. 2020. Predicting ICD-9 codes from medical notes–does the magic of BERT applies here?

Hang Dong, Vctor Suárez-Paniagua, William Whiteley, and Honghan Wu. 2021. Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *Journal of Biomedical Informatics*, 116:103728.

Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. 2020. Dilated convolutional attention network for medical code assignment from clinical text. In *Proceedings of the Third Clinical Natural Language Processing Workshop*, pages 73–78.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-Wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 8180–8187.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1101–1111.

Juri Opitz and Sebastian Burst. 2021. Macro F1 and Macro F1. ArXiv preprint arXiv:1911.03347.

Najmeh Sadoughi, Greg P. Finley, James Fone, Vignesh Murali, Maxim Korenevski, Slava Baryshnikov, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. 2018. Medical code prediction with multi-view convolution and description-regularized label-dependent attention. ArXiv preprint arXiv:1811.01468.

Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P. Xing. 2017. Towards automated ICD coding using deep learning. ArXiv preprint arXiv:1711.04075.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2951–2959.

Fei Teng, Wei Yang, Li Chen, LuFei Huang, and Qiang Xu. 2020. Explainable prediction of medical codes with knowledge graphs. *Frontiers in Bioengineering and Biotechnology*, 8:867.

Shang-Chi Tsai, Ting-Yun Chang, and Yun-Nung Chen. 2019. Leveraging hierarchical category knowledge for data-imbalanced multi-label diagnostic text understanding. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis*, pages 39–43.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for ICD coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3335–3341.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Xiancheng Xie, Yun Xiong, Philip S. Yu, and Yangyong Zhu. 2019. EHR coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 649–658.

A Additional Implementation and Experimental Details

Before a stochastic gradient step on a batch of data, Mullenbach et al. (2018) pad sequences with zeros

so that all documents in this batch have the same number of tokens. Thus results of the forward operation depend on the batch size. This setting causes issues in validation because a result independent of the batch size is needed. Further, for many applications one instance appears at a time in the prediction stage. Thus we follow Mullenbach et al. (2018) to use

$$\text{batch size} = 1$$

in validation and prediction.

After the convolutional layer, Mullenbach et al. (2018) consider the tanh activation function. For both convolutional and linear layers, a bias term is included.

Before the training process, Mullenbach et al. (2018) sort the data according to their lengths. In the stochastic gradient procedure, data are not reshuffled. Therefore, instances considered in each batch are the same across epochs. While this setting is less used in other works, we follow suit to ensure the reproducibility of their results.

In the stochastic gradient procedure, we follow (Mullenbach et al., 2018) to set 200 as the maximal number of epochs. This setting is different from the default 100 epochs in the software LibMultiLabel employed for our experiments. In most situations, the program does not reach the maximal number of epochs. Instead, it terminates after the validation P@5 does not improve in 10 epochs. This criterion also follows from Mullenbach et al. (2018).

All models were trained on one NVIDIA Tesla P40 GPU compatible with the CUDA 10.2 platform and cuDNN 7.6. Note that results may slightly vary if experiments are run on different architectures.

B A Note on Macro-F1

Mullenbach et al. (2018) report macro-F1 defined as

F1 value of macro-precision and macro-recall,

where macro-precision and macro-recall are respectively the mean of precision and recall over all classes. This definition is different from the macro-F1 used in most other works. Specifically, F1 values are obtained for each class first and their mean is considered as Macro-F1; see the discussion of the Macro-F1 definitions in Opitz and Burst (2021). Because works mentioned in Table 2 may not indicate if they use the same Macro-F1 formula as Mullenbach et al. (2018), readers should exercise caution in interpreting Macro-F1 results in Table 2. However, based on Micro-F1 and P@5 results the main point of this paper still stands.

Distinct Label Representations for Few-Shot Text Classification

Sora Ohashi[†], **Junya Takayama**[†], **Tomoyuki Kajiwara**[‡], **Yuki Arase**[†]
[†] Graduate School of Information Science and Technology, Osaka University
[‡] Graduate School of Science and Engineering, Ehime University
[†] {ohashi.sora, takayama.junya, arase}@ist.osaka-u.ac.jp
[‡] kajiwara@cs.ehime-u.ac.jp

Abstract

Few-shot text classification aims to classify inputs whose label has only a few examples. Previous studies overlooked the semantic relevance between label representations. Therefore, they are easily confused by labels that are semantically relevant. To address this problem, we propose a method that generates distinct label representations that embed information specific to each label. Our method is widely applicable to conventional few-shot classification models. Experimental results show that our method significantly improved the performance of few-shot text classification across models and datasets.

1 Introduction

Few-shot text classification (Ye and Ling, 2019; Sun et al., 2019; Gao et al., 2019; Bao et al., 2020) has been actively studied aiming to classify texts whose labels have only a few examples. Such infrequent labels are pervasive in datasets in practice, which are headaches for text classifiers because of the lack of training examples. Snell et al. (2017) showed that the conventional text classifiers are annoyed by the over-fitting problem when the distribution of labels is skewed in a dataset.

Few-shot classification has two approaches: metric-based and meta-learning based methods. The metric-based methods conduct classification based on distances estimated by a certain metric, *e.g.*, cosine similarity (Vinyals et al., 2016), euclidean distance (Snell et al., 2017), convolutional neural networks (Sung et al., 2018), and graph neural networks (Satorras and Estrach, 2018). Metric-based methods in natural language processing focus on representation generation that are suitable for few-shot classification using the attention mechanism with various granularity (Sun et al., 2019), local and global matching of representations (Ye and Ling, 2019), and word co-occurrence patterns in attention mechanisms (Bao et al., 2020). In contrast, meta-larning based methods *learn to learn* for achieving higher accuracy by learning parameter generation (Finn et al., 2017), learning rates and parameter updates (Li et al., 2017; Antoniou et al., 2019), and parameter updates using gradients (Andrychowicz et al., 2016; Ravi and Larochelle, 2017; Li and Malik, 2017).

TECH	Apple confirms it slows down old iPhones as their batteries age Self-driving cars may be coming sooner than you thought
BIZ	Apple apologizes for slowed iPhones, drops price of battery replacements Wall Street isn't too worried about first self-driving Tesla death

Table 1: Examples from Huffpost (BIZ: BUSINESS)

All of these previous studies overlooked the effects of the semantic relevance between label representations, which confuses few-shot classifiers. As a result, the classifiers tend to fail distinguishing examples with semantically relevant labels. Table 1 shows examples with labels sampled from Huffpost (Misra, 2018). The label pair of TECH and BUSINESS is semantically relevant, for which the classifiers are easily confused.

To address this problem, we propose a mechanism that compares label representations to derive distinctive representations. It learns semantic differences between labels and generates representations that embed information specific to each label. Our method can be easily applied to existing few-shot classification models.

We evaluated our method using the standard benchmarks of Huffpost and FewRel (Han et al., 2018). Experimental results showed that our method significantly improved the performance of previous few-shot classifiers across models and datasets, and achieved the state-of-the-art accuracy.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 831–836
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Few-Shot Text Classification

This section describes the problem definition and a general form of conventional few-shot classifiers.

2.1 Problem Definition

In few-shot text classification, sets of supports and queries are given as input. A support set S consists of pairs of text x and corresponding label y: $S = \{(x_i, y_i)|i \in \{1, 2, \cdots, NK\}\}$. N is the number of label types in the support set and K is the number of samples per label type. A query set Q consists of M texts to be classified: $Q = \{q_j|j \in \{1, 2, \cdots, M\}\}$. Note that S and Q have the same set of label types. A few-shot text classifier aims to predict a label for each q_j.

In few-shot classification, training and evaluation are performed on a subset of a dataset called as *episode* (Vinyals et al., 2016). A setting of $N = n$ and $K = k$ is called as n-way k-shot classification. A training episode is created by sampling $k + m$ examples with n types of labels from a training set, and then by dividing them into support and query sets, where $m = \frac{M}{n}$. An evaluation episode is created in the same manner using an evaluation set. Note that labels in the training and evaluation episodes are exclusive, *i.e.*, the classifier is required to predict labels that it has not been exposed during training. The performance of a model is measured using the macro-averaged accuracy of all episodes.

2.2 General Form of Few-shot Text Classification Models

A classification model first converts texts in the support and query sets into vector representations. We denote a subset $S_l \subset S$ as $S_l = \{(x_l^p, y_l^p)|y_l^p = l, p \in \{1, 2, \cdots, K\}\}$ in which all texts have the same label l. An encoder $E(\cdot)$ converts x_l^p and a query q_j to vectors, $\boldsymbol{x}_i^p \in \mathbb{R}^d$ and $\boldsymbol{q}_j \in \mathbb{R}^d$ (d is the dimension of representations), respectively:

$$\boldsymbol{x}_l^p = E\left(x_l^p\right), \quad \boldsymbol{q}_j = E(q_j). \tag{1}$$

$E(\cdot)$ can be any text encoder, such as recurrent neural networks (Yang et al., 2016), convolutional neural networks (Kim, 2014), and pre-trained language models like BERT (Devlin et al., 2019).

Second, the classification model generates a label representation for l. Let $C(\cdot)$ be the function that generates the label representation $\boldsymbol{l} \in \mathbb{R}^d$:

$$\boldsymbol{l} = C\left(\boldsymbol{x}_l^1, \boldsymbol{x}_l^2, \cdots, \boldsymbol{x}_l^K\right). \tag{2}$$

$C(\cdot)$ is typically a pooling function, such as average pooling and max pooling.

Finally, the model calculates the similarity between $\boldsymbol{q}_j$ and each label representation $\boldsymbol{l}_i$ ($i \in \{1, 2, \cdots, N\}$) using a function $R(\cdot)$, and predicts a label whose representation is most similar to that of the query. The probability distribution of the i-th label is computed as:

$$p(i|\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N, \boldsymbol{q}_j) = \frac{e^{R(\boldsymbol{l}_i, \boldsymbol{q}_j)}}{\sum_i e^{R(\boldsymbol{l}_i, \boldsymbol{q}_j)}}. \tag{3}$$

$R(\cdot)$ can be any metrics for estimating similarity. In natural language processing, cosine similarity is a standard choice.

As a loss function L_c, negative log-likelihood is commonly used:

$$L_c = -\frac{1}{M}\sum_{i=1}^{M} \log p(y_j), \tag{4}$$

where y_j is the gold-standard label of q_j.

3 Proposed Method

Figure 1 shows the overview of our method. It adds a mechanism for learning to generate distinctive label representations into conventional few-shot classification models by converting its training into multi-task learning. Our method adds a difference extractor (Section 3.1) and a loss function based on mutual information (Section 3.2) to an arbitrary few-shot classification model.

3.1 Difference Extractor

The difference extractor compares a set of N label representations $\boldsymbol{l}_i$ obtained by Equation (2) with each other and generates representations that retains only the information specific to each label. For doing so, a label representation should depend on a query q_j as classification is conducted based on similarity between the query and labels as shown in Equation (3) (Ye and Ling, 2019). Hence, we model both the label and query representations simultaneously. Specifically, the label representations $\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N$ and the query representation $\boldsymbol{q}_j$ are transformed as:

$$\boldsymbol{H} = \text{MultiHeadAttention}(\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N, \boldsymbol{q}_j), \tag{5}$$

$$\hat{\boldsymbol{l}}_i = \text{GELU}(\boldsymbol{W}_1\boldsymbol{H}_{l_i} + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{6}$$

$$\hat{\boldsymbol{q}}_j = \text{GELU}(\boldsymbol{W}_1\boldsymbol{H}_{q_j} + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{7}$$

where MultiHeadAttention$(\cdot)$ is a self-attention mechanism (Vaswani et al., 2017) that outputs

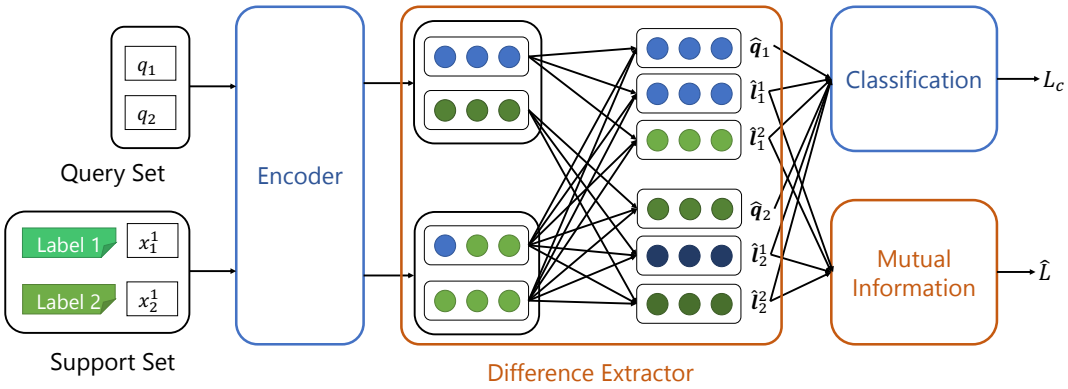


Figure 1: Outline of our method: Components of red boxes are applied to conventional few-shot classifiers.

$\boldsymbol{H} \in \mathbb{R}^{d\times(N+1)}$ hidden representations. $\boldsymbol{H}_{l_i} \in \mathbb{R}^d$ is an output of the self-attention corresponds to l_i, and similarly, $\boldsymbol{H}_{q_j} \in \mathbb{R}^d$ is that of q_j. These hidden representations are further transformed by fully-connected layers with the activation function of GELU$(\cdot)$ (Hendrycks et al., 2020).

3.2 Design of Loss Function

We assume that an ideal representation $\hat{\boldsymbol{l}}_i$ retaining only information specific to an i-th label satisfies that $I(\hat{\boldsymbol{l}}_i; \hat{\boldsymbol{l}}_r) = 0$ for all $\hat{\boldsymbol{l}}_r \ \ (i \neq r)$, where $I(\cdot)$ computes mutual information (MI). That is, each label representation is independent. Hence, we propose an MI-based loss function $\hat{L}$, which constrains such that a label representation $\hat{\boldsymbol{l}}_i$ contains only information specific to the i-th label by minimizing:

$$\hat{L} = \sum_{1\leq i,r\leq N, i\neq r} I(\hat{\boldsymbol{l}}_i, \hat{\boldsymbol{l}}_r). \tag{8}$$

Because the exact value of Equation (8) is difficult to calculate in practice, we minimize its upper-bound following Cheng et al. (2020):

$$\hat{I}(\hat{\boldsymbol{l}}_i; \hat{\boldsymbol{l}}_r) = \sum_{j=1}^{|Q|} R_j, \tag{9}$$

$$R_j = \left[\log p_\theta(\hat{\boldsymbol{l}}_i^j | \hat{\boldsymbol{l}}_r^j) - \frac{1}{|Q|} \sum_{j'=1}^{|Q|} \log p_\theta(\hat{\boldsymbol{l}}_i^j | \hat{\boldsymbol{l}}_r^{j'}) \right],$$

where $p_\theta(\cdot)$ is a neural network which approximates the conditional probability $p(\hat{\boldsymbol{l}}_i^j | \hat{\boldsymbol{l}}_r^j)$.

Finally, the overall loss function is:

$$L = L_c + \alpha \hat{L}, \tag{10}$$

where $\alpha(> 0)$ balances the effect of $\hat{L}$.

4 Experiment

We evaluated our method on different few-shot classification models using the standard benchmarks.

4.1 Benchmark Datasets

Following previous studies (Bao et al., 2020; Gao et al., 2019; Ye and Ling, 2019; Sun et al., 2019), we use Huffpost and FewRel as benchmarks.[1] Following these previous studies, we evaluated the performance of each model using $1,000$ episodes. Because episode generation involves random sampling from a dataset, we repeated this process for 10 times and computed the macro-averaged accuracy as the final score. The statistic significance was measured using a bootstrap significance test.

Huffpost This dataset consists of titles extracted from HuffPost[2] articles. The task is a prediction of a category of an article from its title. The training, validation, and test sets contain 20, 5, and 16 types of labels, respectively. The number of examples per label is 900.

FewRel The task is a prediction of a relation between entities. The training, validation, and test sets contain 65, 5, and 10 types of labels, respectively. The number of examples per label is 700.

4.2 Compared Models and Training Settings

We applied our method on three few-shot classifiers to investigate its effects on different models. As the de-facto standard of metric-based and

[1] Downloaded from https://github.com/YujiaBao/Distributional-Signatures

[2] https://www.huffpost.com/

	Huffpost				FewRel			
	5-Way		10-Way		5-Way		10-Way	
	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
ProtoNet	51.03	68.36	37.42	55.81	78.61	88.92	65.97	80.38
ProtoNet + DE	**51.76**	**69.07**	**38.08**	**56.85**	77.35	88.85	64.96	80.44
ProtoNet + DE + $\hat{L}$	**52.34***	**69.66***	**38.83***	**57.26***	**79.52***	**89.28***	**68.08***	**82.51***
MAML	51.10	65.23	37.37	51.74	68.94	76.49	58.07	65.01
MAML + DE	**51.80**	**67.28**	**38.36**	**53.54**	**75.45**	**85.06**	**62.33**	**72.31**
MAML + DE + $\hat{L}$	**51.71**	**67.38**	**38.11**	**53.75***	**75.99***	**84.07**	**63.13***	**70.99**
MLMAN	47.07	57.80	33.86	43.79	73.61	82.75	60.28	71.48
MLMAN + DE	**49.73**	**60.94**	**36.37**	**47.25**	**74.38**	**83.67**	**61.14**	**72.70**
MLMAN + DE + $\hat{L}$	**48.98**	**60.75**	**35.60**	**46.73**	**78.21***	**86.43***	**65.44***	**76.43***
Bao et al. (2020)	42.12	62.97	-	-	70.08	88.07	-	-

Table 2: Experimental results (**bold** and * indicate significantly higher accuracies compared to each baseline model and baseline + DE, respectively.)

meta-learning based models, we employed ProtoNet (Snell et al., 2017) and MAML (Finn et al., 2017), respectively. Besides, we employed MLMAN (Ye and Ling, 2019), which is the state-of-the-art few-shot classification model on FewRel. We also compared to Bao et al. (2020), which achieved the sate-of-the-art on HuffPost.

As the Encoder $E(\cdot)$ and pooling function $C(\cdot)$ for each model, we used the BERT-base, uncased[3] and average pooling, respectively, which showed strong performance in various text classification tasks (Devlin et al., 2019). We used PyTorch and Huggingface Transformers (Wolf et al., 2020) for implementation.[4]

We applied our difference extractor and MI-loss function (denoted as "+ DE + $\hat{L}$") to ProtoNet, MAML, and MLMAN. For the difference extractor, we used 1-layer self-attention mechanism with 8-heads. As an ablation study, we also compared our method that only applies the difference extractor (denoted as "+ DE"), which is trained only with the classification loss (Equation (4)).

We trained all models with 5-way 1-shot setting. We then tested the models on different ways and shots. As an optimizer, we used Adam (Kingma and Ba, 2015). A learning rate and α in Equation (10) were searched in ranges of $[1e-5, 3e-5, 5e-5]$ and $[1e-6, 1e-4, 1e-2, 1]$, respectively, to maximise accuracy on the validation set.

[3] https://github.com/google-research/bert

[4] Our code is available at https://github.com/21335732529sky/difference_extractor

4.3 Overall Results

As Table 2 shows, our method significantly improved all of the baseline models across datasets.[5] For MAML and MLMAN, our difference extractor always improved the performance of the original models. By combination with the MI-loss, the performance improved by from 0.61 up to 7.68 points. In contrast, applying only the difference extractor to ProtoNet, *i.e.*, ProtoNet + DE, deteriorated its original performance on FewRel dataset. These results confirm that both the difference extractor and MI-loss are crucial for ProtoNet. By using both, ProtoNet + DE + $\hat{L}$ consistently improved the baseline by from 0.39 up to 2.13 points.

4.4 Impact of DE and MI-loss on Baselines

The experimental results confirmed that the combination of our difference extractor and MI-loss function consistently improved the few-shot classification models. In particular, MI loss is more effective for a simpler model, *i.e.*, ProtoNet. MLMAN has an internal mechanism for comparing supports and queries, and MAML has a mechanism for updating the model parameters to accurately classify supports. These internal mechanisms allow to learn label representations that boost classification accuracy. Hence, the functionality of MI loss is partly achieved by these internal mechanisms. On the other hand, ProtoNet has the simplest architec-

[5] Note that the performance of ProtoNet was higher than that in (Ye and Ling, 2019) and (Bao et al., 2020). This is because we tuned the learning rate using the development set.

	Huffpost	FewRel
ProtoNet + DE + $\hat{L}$	$1e-4$	$1e-4$
MLMAN + DE + $\hat{L}$	$1e-4$	$1e-2$
MAML + DE + $\hat{L}$	$1e-6$	$1e-4$

Table 3: Weight of MI loss determined to maximise the performance on the development set

ture as described in Section 2.2 without additional mechanisms. Hence, both of the difference extractor and our loss function are crucial for ProtoNet.

Another factor affecting the performance of MI loss is the number of labels in a datset. When the number of labels is large, semantically relevant labels more likely exist, where MI loss plays a role. This assumption was empirically confirmed by the fact that FewRel, where MI loss (DE + $\hat{L}$) outperformed DE for most cases, has 80 labels. On the other hand, Huffpost has about half the number of labels (i.e., 41 labels).

4.5 Impact of Hyperparameters

Table 3 shows the settings of α tuned on the development set. Overall, the values of α on FewRel are larger than those on Huffpost. Larger α values increase the influence of MI loss on models, which is effective on datasets with a large number of labels like FewRel.

Figure 2 shows the accuracy measured on the development set when varying α. The performance tends to decrease when α is set too large. We suspect that too large α forces models to extract differences irrelevant to the classification task. For example, the second examples in Table 1 are about self-driving cars, where only the `BIZ` example contains named entities of `Wall Street` and `Tesla`. It is a noticeable difference; however, unlikely be useful for the classification task. Label representations of such spurious distinctiveness may degrade the classification performance.

5 Conclusion and Future Work

In this paper, we introduced a novel method shedding light on semantic relations between labels. Our method improved the classification accuracy of representative few-shot classifiers on both Huffpost and FewRel datasets, confirming the reasonable applicability of the proposed method.

Technically, our method can be applied to other classification problems that handle semantic labels,

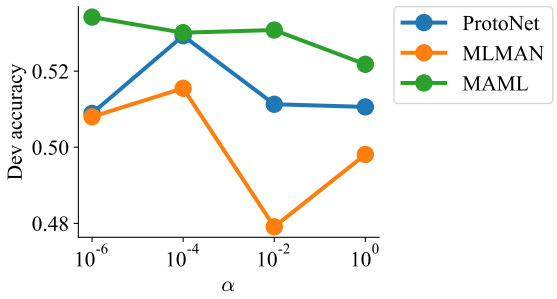


Figure 2: Accuracy on the Huffpost development set when varying α values

such as image and entity classifications. We will conduct evaluation to see its effects on various types of classifcations.

Acknowledgments

This work was supported by JST AIP-PRISM Grant Number JPMJCR18Y1, Japan.

References

Marcin Andrychowicz, Misha Denil, Sergio Gómez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. 2016. Learning to Learn by Gradient Descent by Gradient Descent. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pages 3981–3989.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2019. How to Train Your MAML. In *Proceedings of the 7th International Conference on Learning Representations*, pages 1–11.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot Text Classification with Distributional Signatures. In *Proceedings of the 8th International Conference on Learning Representations*, pages 1–24.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 6407–6414.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained Transformers Improve Out-of-Distribution Robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15.

Ke Li and Jitendra Malik. 2017. Learning to Optimize. In *Proceedings of the 5th International Conference on Learning Representations*, pages 1–13.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *arXiv:1707.09835*, pages 1–11.

Rishabh Misra. 2018. News Category Dataset.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-Shot Learning. In *Proceedings of the 5th International Conference on Learning Representations*, pages 1–11.

Victor Garcia Satorras and Joan Bruna Estrach. 2018. Few-Shot Learning with Graph Neural Networks. In *Proceedings of the 6th International Conference on Learning Representations*, pages 1–13.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 4077–4087.

Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical Attention Prototypical Networks for Few-Shot Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 476–485.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 5998–6008.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, pages 3630–3638.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2872–2881.

Learning to Solve NLP Tasks in an Incremental Number of Languages

Giuseppe Castellucci
Amazon
Seattle, USA
giusecas@amazon.com

Simone Filice
Amazon
Tel Aviv, Israel
filicesf@amazon.com

Danilo Croce
Dept. of Enterprise Engineering
University of Rome, Tor Vergata
Roma, Italy
croce@info.uniroma2.it

Roberto Basili
Dept. of Enterprise Engineering
University of Rome, Tor Vergata
Roma, Italy
basili@info.uniroma2.it

Abstract

In real scenarios, a multilingual model trained to solve NLP tasks on a set of languages can be required to support new languages over time. Unfortunately, the straightforward retraining on a dataset containing annotated examples for all the languages is both expensive and time-consuming, especially when the number of considered languages grows. Moreover, the original annotated material may no longer be available due to storage or business constraints. Re-training only with the new language data will inevitably result in Catastrophic Forgetting of previously acquired knowledge. We propose a Continual Learning strategy that updates a model to support new languages over time, while maintaining consistent results on previously learned languages. We define a Teacher-Student framework where the existing model "teaches" to a student model its knowledge about the languages it supports, while the student is also trained on a new language. We report an experimental evaluation in several tasks including Sentence Classification, Relational Learning and Sequence Labeling.

1 Introduction

In Natural Language Processing (NLP), multilingualism refers to the capability of a single model to cope with multiple languages. Recently, different Transformer-based architectures have been extended to operate over multiple languages, as in Conneau et al. (2020); Conneau and Lample (2019); Pires et al. (2019). Despite these models can be applied in the zero-shot setting (Xian et al., 2019; Artetxe and Schwenk, 2019), in many practical applications their quality will not be satisfactory. Instead, fine-tuning over annotated material in each target language is needed to obtain competitive results, as the experimental results in Lewis et al. (2019); Tran and Bisazza (2019) suggest. Having annotated material for all the languages is not always possible, especially when the model has to support an incremental number of new languages over time. In fact, the original fine-tuning material may no longer be available for storage, business or privacy constraints. For example, in a real-world application, customers may request deletion of their data, or the service itself may provide specific data retention policies, or the adopted model may be provided by a third party that did not release the training data (Chen and Moschitti, 2019). In these cases, new language support can be added in a Continual Learning (CL) setting (Lange et al., 2019), that is fine-tuning the model only using the annotated material for the new language(s). However, this approach is vulnerable to the *Catastrophic Forgetting* (CF) (McCloskey and Cohen, 1989) of previously learned languages, a well-documented concern discussed in Chen et al. (2018): when a model is incrementally fine-tuned on new data distributions, it risks forgetting how to treat instances of the previously learned ones.

In this paper, we propose a CL strategy for updating a model over an incremental number of languages, so that at each step the model requires only annotated examples of the new language(s). Our goal is to remove the dependency on the original fine-tuning material and reduce the need for annotated data at each training step. We propose a Teacher-Student framework inspired by the *Knowledge Distillation* (KD) literature (Hinton et al., 2015). Although this technique is traditionally used for the purpose of model compression (Sanh et al., 2019), recent works in Computer Vision applied KD to incrementally learn image processing tasks (Li and Hoiem, 2018). Here, we adopt KD to miti-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 837–847
August 1–6, 2021. ©2021 Association for Computational Linguistics

gate CF when incrementally training Transformer-based architectures (Devlin et al., 2019) for semantic processing tasks. The existing model (here the teacher) imparts knowledge to a (student) model about the languages it already supports, while this is trained on new languages.

We evaluated our approach using multilingual BERT-based models on three semantic processing tasks, involving Sentence Classification, Paraphrase Identification and Sequence Tagging. Results suggest that the model can progressively learn new languages, while maintaining or even improving its quality over previously observed ones.

2 Related Work

Continual Learning (CL) (Chen et al., 2018) studies how to train a machine from a stream of data, which can evolve over time by changing the input distribution or by incorporating new tasks. CL aims to gradually extend the knowledge in a model (Lange et al., 2019), while avoiding Catastrophic Forgetting (Goodfellow et al., 2013). Previous work has mostly focused on Computer Vision (Shmelkov et al., 2017; Li and Hoiem, 2018; Rannen et al., 2017) by using Knowledge Distillation (KD) (Hinton et al., 2015) as the base framework.

CL in NLP, as opposed to Computer Vision, is still nascent (Greco et al., 2019; Sun et al., 2020). This reflects in the small number of proposed methods to alleviate CF, as discussed in Biesialska et al. (2020). In this context, some works focus on the Online Learning aspect of the CL (Filice et al., 2014). In NLP, KD has been mainly adopted to compress models (Kim and Rush, 2016; Sanh et al., 2019), and was only recently applied for CL in Named Entity Recognition (Monaikul et al., 2021).

In the context of multilingual analysis, most of the works leverage Domain Adaptation techniques within Machine Translation (Dong et al., 2015; Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017; Tan et al., 2019) in order to apply a machine translation model to an increasing set of languages.

To the best of our knowledge, this is the first work adopting CL to mitigate CF when training Transformer-based models in an incremental number of languages for semantic processing tasks.

3 CL for Multilingual processing

Multilingual Continual Learning. In the targeted scenario, we have a multilingual neural model, namely $\mathcal{M}_{L_A}$, originally pre-trained on a set of languages $L_P = \{l_1, l_2, \ldots\}$ (such as multilingual BERT (Pires et al., 2019)) and already fine-tuned to solve a task $\mathcal{T}$ (such as sentence classification) on a given set of languages $L_A \subset L_P$. The scope is to extend such model to solve $\mathcal{T}$ on a set of new languages $L_B \subset L_P$, with $L_A \cap L_B = \emptyset$.

In the rest of the discussion, without loss of generality, we assume that $L_B = \{l_{new}\}$, i.e., we support only one new language at a time. In case $n > 1$ new languages need to be added, a sequence of n model extensions can be performed. In our setting, we assume that: (*i*) a new annotated dataset $S_{\{l_{new}\}}$ for task $\mathcal{T}$ in language l_{new} is available; (*ii*) the examples used to fine-tune $\mathcal{M}_{L_A}$ are not available anymore; (*iii*) unlabeled examples are available in each language from L_A. Since $l_{new} \in L_P$, i.e., the original pre-training stage included l_{new}, the model could already operate in a zero-shot setting (i.e., without any fine-tuning stage involving l_{new} data). However, the performance of the zero-shot setting is typically non-satisfactory and a dedicated fine-tuning on l_{new} is generally required. A *naive* CL strategy consists of fine-tuning $\mathcal{M}_{L_A}$ over $S_{\{l_{new}\}}$. However, even though this schema is supposed to produce an effective model for l_{new} instances, it is not guaranteed that the resulting model would still be competitive on languages L_A, due to CF (Greco et al., 2019; Sun et al., 2020). An alternative greedy solution consists of adopting *self-training* as in Rosenberg et al. (2005): $\mathcal{M}_{L_A}$ is used to annotate some unlabeled examples in languages L_A so that the resulting pseudo-labeled dataset $\tilde{S}_{L_A}$ can be used together with $S_{\{l_{new}\}}$ to fine-tune $\mathcal{M}_{L_A}$ and mitigate CF. Unfortunately, this can also reinforce the errors of $\mathcal{M}_{L_A}$, as discussed in Hinton et al. (2015).

Preventing Catastrophic Forgetting. CF is typically caused by the model's weights, which are pushed towards fitting the data of the latest fine-tuning stage. If the model is not trained using examples in languages L_A, it risks forgetting how to treat them. To overcome CF, we propose a method based on Knowledge Distillation (KD). We define a Teacher-Student framework where $\mathcal{M}_{L_A}$ acts as the teacher, while the student is a clone of $\mathcal{M}_{L_A}$ which is fine-tuned using the multi-loss function $\mathcal{L}_{CL} = \mathcal{L}_{\mathcal{T}} + \mathcal{L}_{KD}$. The term $\mathcal{L}_{\mathcal{T}}$ is the task-specific loss, computed on the annotated examples from $S_{\{l_{new}\}}$. $\mathcal{L}_{KD}$ is a distillation loss computed on U_{L_A}, a set of unlabeled examples written in the previous languages L_A and here processed by the

teacher model. $\mathcal{L}_{\mathcal{T}}$ thus pushes the model to learn how to solve $\mathcal{T}$ in the new language l_{new}. $\mathcal{L}_{KD}$ helps the model maintaining a consistent performance on the languages L_A, by forcing the student to mimic the teacher predictions on data resembling the data distribution observed in L_A. In order to define $\mathcal{L}_{KD}$ consistently with Hinton et al. (2015), let us define $d_i(x)$ as the output logits of the model's last layer when applied to an example x. The logits are converted into a class-probability distribution using the temperature-softmax:

$$y_i(x) = \frac{exp(d_i/T)}{\sum_j exp(d_j/T)}$$

where T is a temperature hyper-parameter, which controls the smoothness of the distribution. $\mathcal{L}_{KD}$ is thus computed as the cross-entropy between the output probability distributions provided by the student and teacher, namely y_i^s and y_i^t, i.e.:

$$\mathcal{L}_{KD}(x) = -\sum_i y_i^t(x) \log y_i^s(x)$$

Using $\mathcal{L}_{KD}$ instead of the self-training procedure preserves the uncertainty of the teacher's model and prevents the student from amplifying the teacher's errors, as demonstrated in Hinton et al. (2015).

4 Experimental Evaluation

This section presents the results of the proposed CL strategy over three semantic processing tasks, involving text classification and sequence tagging. In particular, we report the Mean Absolute Error (MAE) over the Multilingual Amazon Review Corpus (MARC) (Keung et al., 2020), i.e., a 5 category Sentiment Analysis task in 6 languages. We report the Accuracy over a sentence-pair classification task, i.e., Paraphrase Identification on the PAWS-X dataset (Yang et al., 2019) in 6 languages[1]. Finally, we report the F1 for the Named Entity Recognition (NER) in 4 languages by merging the CoNLL 2002 (Tjong Kim Sang, 2002) and 2003 (Tjong Kim Sang and De Meulder, 2003) datasets. Additional details about the datasets are in Appendix.

Experimental Setup. We foresee a setting where a BERT-based model is incrementally trained using annotated datasets in multiple languages. At each step, the model is fine-tuned using a dataset in one specific language, while the annotated material used up to that point is discarded.

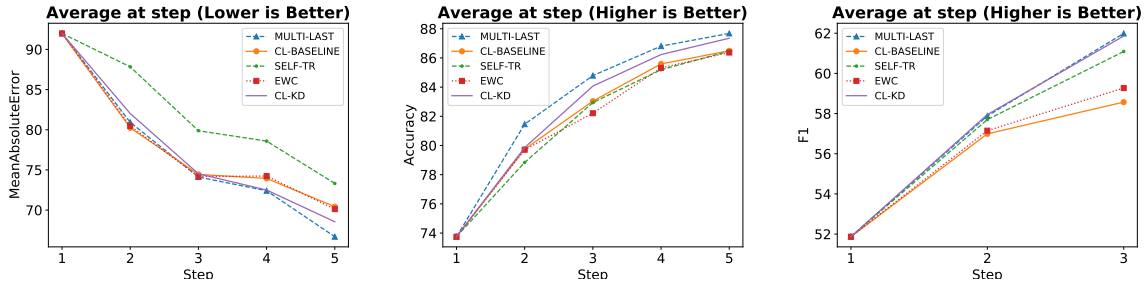
We reasonably assume that a set of unlabeled data is available for the languages already observed. In order to simulate this scenario, we designed a data splitting procedure such that each annotated example is observed only in one step. Let us assume we observe languages in the order $l_1 \rightarrow, \ldots, \rightarrow l_n$. For each language l_i, its training set $D_{\{l_i\}}$ is divided into $n-i+1$ equal slices, i.e., $(D_{\{l_i\}}^{(i)}, \ldots, D_{\{l_i\}}^{(n-i+1)})$. Depending on the learning strategy, each slice will be either annotated (indicated with a S symbol) or not annotated (indicated with a U symbol). At the last step, we will have observed all the data, either annotated or not.

Learning Strategies. We compare four CL strategies. We denote with `CL-Baseline` the strategy where at step k the model $\mathcal{M}_k$ is obtained by updating $\mathcal{M}_{k-1}$ by using only the $S_k = S_{\{l_k\}}^{(1)}$ annotated dataset, only with the task loss $\mathcal{L}_T$. The second strategy is denoted with `Self-Training`: at step k, $\mathcal{M}_{k-1}$ is used to annotate the dataset $\tilde{S}_k = \bigcup_{j=1}^{k-1} \{U_{\{l_j\}}^{(k-j+1)}\}$. $\mathcal{M}_k$ is then fine-tuned by using $S_k = S_{\{l_k\}}^{(1)} \cup \tilde{S}_k$ with the task loss $\mathcal{L}_T$. We denote with `CL-KD` the strategy we propose, where at step k, $\mathcal{M}_{k-1}$ is used as the teacher in our proposed KD schema[2]. $\mathcal{M}_{k-1}$ is used to derive the target output distribution of the dataset $U_k = \bigcup_{j=1}^{k-1} \{U_{\{l_j\}}^{(k-j+1)}\}$. $\mathcal{M}_k$ is then trained by adopting $S_k = S_{\{l_k\}}^{(1)}$ with the task loss $\mathcal{L}_T$ and U_k with the loss $\mathcal{L}_{KD}$. We compared with a further competitive method, namely Elastic Weight Consolidation, here denoted with `EWC` (Kirkpatrick et al., 2017). This popular CL procedure applies a regularization technique that penalizes large variations on those model's weights that are the most important for the tasks learned so far.

As a sort of upper-bound, we report the results by adopting a non-Continual Learning strategy, i.e., `Multi-Last`, where the model is trained from scratch using an annotated dataset in all languages we want to support at step k. More formally, at step k the data is $S_k = \bigcup_{j=1}^{k} \{S_{\{l_j\}}^{(k-j+1)}\}$, i.e., the annotated data is about k times larger than the one used in the CL settings.

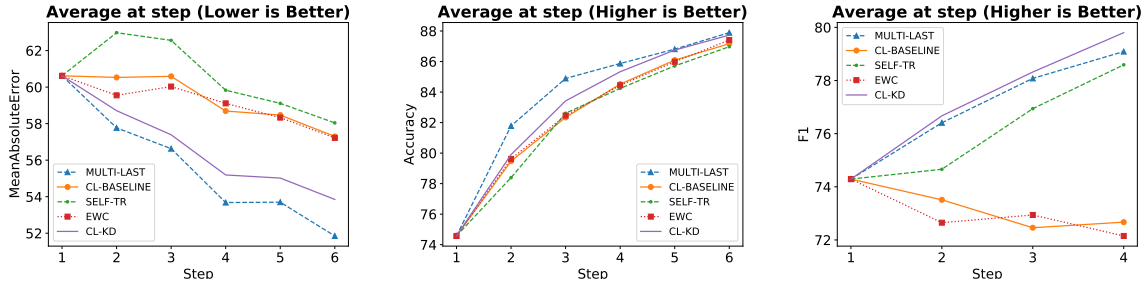
[1] PAWS-X contains 7 languages. We were not able to reproduce the results of Yang et al. (2019) for the Korean language. Thus, we removed this language in our evaluation.

[2] We also investigated an approach inspired by Gururangan et al. (2020): we augmented `CL-KD` with Masked Language Modeling and Next Sentence Prediction objectives to continue the pre-training. Preliminary experiments provided negligible improvements, not reported here due to lack of space.



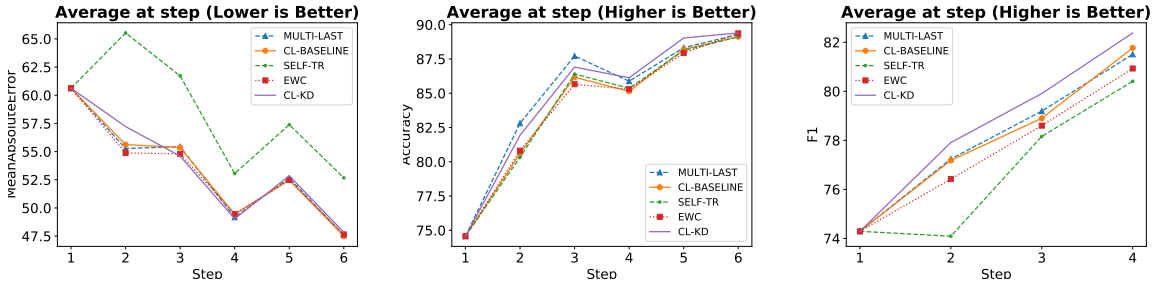
(a) MARC (b) PAWS-X (c) ConLL 2002/2003

Figure 1: Average performance measures for the MARC, PAWS-X and CoNLL for the languages not yet used in training. At each step k, we report the average score for the languages that will be observed in steps $(k+1, \ldots, n)$.



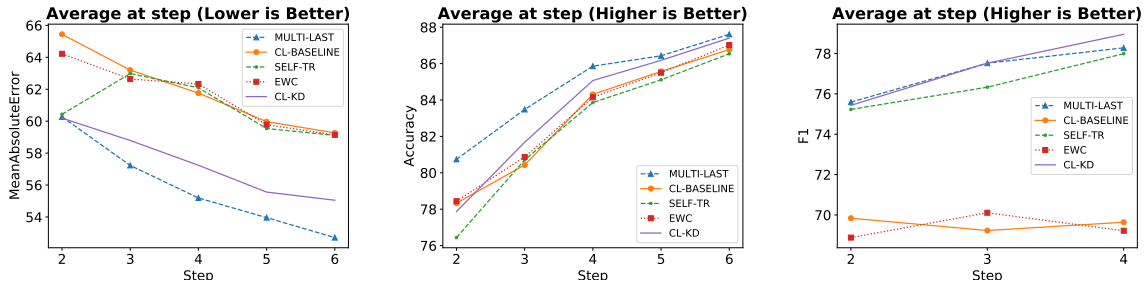
(a) MARC (b) PAWS-X (c) ConLL 2002/2003

Figure 2: Average performance measures for the MARC, PAWS-X and CoNLL. At each step k, we report the average score with respect to the languages observed in steps $(1, \ldots, k)$.



(a) MARC (b) PAWS-X (c) ConLL 2002/2003

Figure 3: Average performance measures on MARC, PAWS-X and CoNLL for the language observed at step k.



(a) MARC (b) PAWS-X (c) ConLL 2002/2003

Figure 4: Average performance measures for the MARC, PAWS-X and CoNLL for the languages observed in the past steps. At each step k, we report the average score with respect to the languages observed in steps $(1, \ldots, k-1)$.

Model Training. We used the *bert-base-multilingual-cased* model in the Huggingface Transformers package (Wolf et al., 2019). We trained the models for 10 epochs with Early Stopping (patience= 3) and batch size 32. After initial experiments, we set the temperature T to 1. We repeated our experiments for 6/6/24 sequences of language permutations for MARC/PAWS-X/CoNLL, and we report the average performances.

Experimental Results and Discussion. We first run zero-shot experiments by fine-tuning a model on a subset of languages and testing it on the unobserved ones (see Figure 1). By comparing the results with the ones in Figure 2, we can observe a large gap between the results achieved on the languages still to be observed vs. the training ones. For instance, at step 1 the average gap is more than 30 MAE on MARC, about 0.8% Accuracy on PAWS-X and about 22 F1 on CoNLL. This confirms the need to fine-tune the model on each language of interest.

Figures 2a, 2b and 2c show the results on MARC, PAWS-X and CoNLL, respectively. At each step, we report the average measure computed over all the observed languages, averaged over all the permutations. Given that we are solving the same task in multiple languages, regardless the adopted strategy, the performance can improve at each step due to a cross-lingual transfer learning effect. This beneficial impact is contrasted by the CF, which is also supposed to increase at each step. In our experiments, the effect of transfer learning is generally stronger, with the only exception of `CL-Baseline` in CoNLL, where CF seems to dominate (the F1 drops from 74.29 at step 1 to 72.67 at step 4). In MARC and PAWS-X, this is alleviated: we argue that CoNLL is more challenging, as it is a word-level tagging on a smaller dataset.

The approach we propose, i.e., `CL-KD`, is able to constantly outperform its corresponding baseline `CL-Baseline`. The adoption of knowledge from the previously encountered languages is crucial in mitigating the CF phenomenon. For example, in MARC the MAE in the `CL-KD` setting is reduced from 60.62 in the first step to 53.85 in the last step. The same applies for PAWS-X where accuracy jumps from 74.57 to 87.73 and for CoNLL with F1 from 74.29 to 79.80. The performances of `CL-KD` are similar to the `Multi-Last` even if this clearly has an advantage, using a larger dataset consisting of examples written in all languages.

Figure 3 reports the average performance on the language observed during the last step only, while Figure 4 shows results on the previously acquired languages. Notice that `CL-KD` achieves comparable results between the previously acquired languages and the last learned one. Conversely, the other CL models perform significantly lower.

Notice that the `CL-KD` model achieves better results than `Self-Training`, especially for MARC and CoNLL. This means that classifying the examples with the previous model amplifies the errors of that model. In PAWS-X, the improvements achieved by `CL-KD` are less evident: we argue this is due to the nature of the dataset, where the training set in each language is derived via automatic machine translation. In any case, `CL-KD` is still performing better than `Self-Training` and `CL-Baseline`: despite automatic translation can be a viable solution, its performances will likely be sub-optimal. Notice that `EWC` is considered one of the most effective approaches for CL, but interestingly in our setting its results are not satisfactory. We investigated if the order of the languages provides significant differences. We did not notice major variations, also when the involved languages are very different[3].

Finally, we trained a full-multilingual model with all the data for all the languages. The `CL-KD` performances are not far from this model, as the difference is only 4.47, 1.56 and 2.44 for MARC, PAWS-X and CoNLL, respectively.

5 Conclusions

This paper investigated a Continual Learning strategy, based on Knowledge Distillation, for training Transformer architectures in an incremental number of languages. We demonstrated that with our approach the model maintains its robustness in processing already acquired languages without having access to annotated data for them, while learning new languages. Future work will apply our methodology to other NLP tasks, such as QA.

Acknowledgments

We would like to thank the "Istituto di Analisi dei Sistemi ed Informatica - Antonio Ruberti" (IASI) for supporting the experimentations through access to dedicated computing resources.

[3]For example, in PAWS-X when *ja* and *zh* are the first two languages, the Accuracy at the last step is 87.53. When *ja* is the third and *zh* is the fifth, the Accuracy is 87.76. Similar outcomes can be observed for the MARC dataset.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lingzhen Chen and Alessandro Moschitti. 2019. Transfer learning for sequence labeling using source model and target data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6260–6267.

Zhiyuan Chen, Bing Liu, Ronald Brachman, Peter Stone, and Francesca Rossi. 2018. *Lifelong Machine Learning*, 2nd edition. Morgan, Claypool Publishers.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Simone Filice, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2014. Effective kernelized online learning in language processing tasks. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, volume 8416 of *Lecture Notes in Computer Science*, pages 347–358. Springer.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks.

Claudio Greco, Barbara Plank, Raquel Fernández, and Raffaella Bernardi. 2019. Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3601–3605, Florence, Italy. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Z. Li and D. Hoiem. 2018. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947.

Michael McCloskey and Neil J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169.

Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, February 2-9, 2021*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Amal Rannen, Rahaf Aljundi, Matthew B. Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *The IEEE International Conference on Computer Vision (ICCV)*.

C. Rosenberg, M. Hebert, and H. Schneiderman. 2005. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pages 29–36.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

K. Shmelkov, C. Schmid, and K. Alahari. 2017. Incremental learning of object detectors without catastrophic forgetting. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3420–3429.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. {LAMAL}: {LA}nguage modeling is all you need for lifelong language learning. In *International Conference on Learning Representations*.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, page 142–147.

Ke M. Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 281–288. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2251–2265.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Datasets

Sentence Classification. We used the Multilingual Amazon Reviews Corpus (MARC) (Keung et al., 2020), i.e., a Sentiment Analysis dataset. MARC is a large-scale collection of Amazon reviews in 6 languages (English, German, Spanish, French, Japanese and Chinese). The dataset is made of $200,000/5,000/5,000$ reviews for each language, respectively for train, validation and test. We refer to the *fine-grained* classification (the target category is on 1-5 scale) by using the *body* of the review.

Sentence-Pairs Classification. We adopted the PAWS-X dataset (Yang et al., 2019) for the Paraphrase Identification task. The dataset is composed of about $24,000$ human translated evaluation pairs and about $296,000$ machine translated training pairs over 7 languages: English, Spanish, French, German, Japanese, Chinese, Korean. We actually didn't used the Korean languages, as in preliminary experiment we were not able to reproduce the results of the (Yang et al., 2019) paper. We suspect a problem in the encoding affected our results in this language with the bert multilingual model.

Sequence Tagging. We reported experiments on Named Entity Recognition (NER) using the CoNLL 2002 (Tjong Kim Sang, 2002) and CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) datasets. We merged the two datasets as in Rahimi et al. (2019) to obtain a single dataset over 4 languages, i.e., English, Spanish, German and Dutch. The dataset contains $51,821/11,344/13,556$ annotated sentences, respectively for train, validation and test. Each sentence has been annotated with respect to the following entities: *Person*, *Location*, *Organization* and *Miscellaneous*.

A.2 Additional Results

In this section we report more details on the results of the experiments already discussed in Section 4.

A.2.1 Results on Observed Languages

Tables 1, 2 and 3 complement the results already shown in Figure 2 and summarizes the average performance on the languages observed till each step for MARC, PAWS-X and ConNLL respectively.

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	60.62	60.62	60.62	60.62	60.62
2	57.77	60.53	62.98	59.55	58.71
3	56.63	60.59	62.56	60.03	57.39
4	53.68	58.69	59.83	59.11	55.19
5	53.70	58.47	59.11	58.33	55.02
6	51.85	57.30	58.04	57.22	53.85

Table 1: MARC performances for the observed languages (as in Figure 2a), i.e., at each step we report the average of the measure for the languages observed including the last step (step $\leq k$). The reported measure is the Mean Absolute Error (lower is better).

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	74.57	74.57	74.57	74.57	74.57
2	81.78	79.47	78.39	79.62	79.90
3	84.89	82.34	82.61	82.46	83.42
4	85.87	84.52	84.24	84.44	85.33
5	86.81	86.09	85.71	85.98	86.75
6	87.89	87.17	86.97	87.41	87.73

Table 2: PAWS-X performances for the observed languages (as in Figure 2b), i.e., at each step we report the average of the measure for the languages observed including the last step (step $\leq k$). The reported measure is the Accuracy (higher is better).

A.2.2 Results on New Language Only

The following results show how an already fine-tuned model learn to manage a new language. While results in Figure 2 are averaged across all languages (observed up to the k-th step) the following evaluations focus

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	74.29	74.29	74.29	74.29	74.29
2	76.41	73.51	74.66	72.65	76.67
3	78.08	72.46	76.94	72.94	78.32
4	79.09	72.67	78.59	72.15	79.80

Table 3: CoNLL 2002/2003 performances for the observed languages (as in Figure 2c), i.e., at each step we report the average of the measure for the languages observed including the last step (step $\leq k$). The reported measure is the F1 (higher is better).

only on the last observed language. Figure 3 and Tables 4, 5 and 6 report the average performance on the last learned language. The average performance tends to improve at each step thanks to the cross-lingual transfer learning effect. All the models perform similarly, exception for the `Self-Training` model that exhibits generally lower results. This is probably due to the error amplification issue that somehow degrades the cross-lingual transfer.

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	60.62	60.62	60.62	60.62	60.62
2	55.28	55.62	65.53	54.88	57.22
3	55.44	55.37	61.72	54.80	54.60
4	49.17	49.45	53.05	49.45	49.04
5	52.65	52.47	57.39	52.49	52.85
6	47.58	47.49	52.67	47.65	47.85

Table 4: MARC performances for the Current Language (as in Figure 3a). At each step we report the measure for the language observed in that step (step $= k$). The reported measure is the Mean Absolute Error (lower is better).

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	74.57	74.57	74.57	74.57	74.57
2	82.81	80.60	80.33	80.81	81.93
3	87.72	86.17	86.40	85.66	86.91
4	85.88	85.14	85.37	85.30	86.13
5	88.32	88.20	88.11	87.93	89.03
6	89.29	89.13	89.12	89.37	89.40

Table 5: PAWS-X performances for the Current Language (as in Figure 3b). At each step we report the measure for the language observed in that step (step $= k$). The reported measure is the Accuracy (higher is better).

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	74.29	74.29	74.29	74.29	74.29
2	77.24	77.18	74.09	76.42	77.91
3	79.19	78.90	78.16	78.60	79.91
4	81.51	81.77	80.41	80.93	82.38

Table 6: CoNLL 2002/2003 performances for the Current Language (as in Figure 3c). At each step we report the measure for the language observed in that step (step $= k$). The reported measure is the F1 (higher is better).

A.2.3 Results on Previously Learned Languages

Figure 4 and Tables 7, 8 and 9 report the average performance for each step on the previously acquired languages. This allows us to better assess the impact of Catastrophic Forgetting. In particular, if we compare these results with the ones reported in Section A.2.2, it is possible to appreciate that model `CL-KD` achieves comparable results between the previously acquired languages and the last learned one. Conversely, the other CL models, and in particular `CL-Baseline`, provide significantly lower results on the previously acquired languages w.r.t. to the language learned during the last training step. This is clearly demonstrating the impact of the Catastrophic Forgetting effect.

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	-	-	-	-	-
2	60.27	65.44	60.43	64.22	60.19
3	57.23	63.20	62.98	62.64	58.79
4	55.19	61.76	62.09	62.33	57.24
5	53.96	59.97	59.54	59.78	55.56
6	52.71	59.27	59.11	59.14	55.05

Table 7: MARC performances for the Past Languages (as in Figure 4a), i.e., at each step we report the average measure for the languages observed till that step (step $< k$). The reported measure is the Mean Absolute Error (lower is better).

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	-	-	-	-	-
2	80.74	78.33	76.44	78.44	77.87
3	83.48	80.43	80.71	80.87	81.68
4	85.86	84.31	83.86	84.15	85.07
5	86.43	85.57	85.11	85.50	86.18
6	87.61	86.77	86.54	87.02	87.40

Table 8: PAWS-X performances for the Past Languages (as in Figure 4b), i.e., at each step we report the average measure for the languages observed till that step (step $< k$). The reported measure is the Accuracy (higher is better).

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	-	-	-	-	-
2	75.59	69.84	75.23	68.88	75.43
3	77.52	69.23	76.33	70.11	77.52
4	78.28	69.64	77.99	69.22	78.95

Table 9: CoNLL 2002/2003 performances for the Past Languages (as in Figure 4c), i.e., at each step we report the average measure for the languages observed till that step (step $< k$). The reported measure is the F1 (higher is better).

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	91.99	91.99	91.99	91.99	91.99
2	80.93	80.24	87.83	80.48	82.02
3	74.12	74.44	79.88	74.18	74.52
4	72.41	73.94	78.59	74.24	72.50
5	66.69	70.43	73.32	70.12	68.55
6	-	-	-	-	-

Table 10: MARC performances for the Future Languages (zero-shot setting, as in Figure 1a). At each step we report the average of the measure for the languages still not observed (step $> k$). The reported measure is the Mean Absolute Error (lower is better).

A.2.4 Results on Untrained Languages

Figure 1 and Tables 10, 11 and 12 report the average performance for each step on the languages that the model did not train on so far.

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	73.75	73.75	73.75	73.75	73.75
2	81.46	79.75	78.84	79.71	79.86
3	84.79	83.04	82.94	82.21	84.07
4	86.81	85.59	85.19	85.32	86.23
5	87.68	86.49	86.47	86.37	87.35
6	-	-	-	-	-

Table 11: PAWS-X performances for the Future Languages (zero-shot setting, as in Figure 1b). At each step we report the average of the measure for the languages still not observed (step $> k$). The reported measure is the Accuracy (higher is better).

This allows us to evaluate the performance of the zero-shot setting. As expected, results are pretty poor,

and the gap between the results on training languages and the zero-shot languages is very large: the gap is more than 30 MAE on MARC, about 8% Accuracy on PAWS-X and about 22 F1 on CoNLL. This confirms the need to fine-tune the model on each language of interest.

Step/Model	MULTI-LAST	CL-BASELINE	SELF-TR	EWC	CL-KD
1	51.87	51.87	51.87	51.87	51.87
2	57.88	56.99	57.70	57.15	57.95
3	61.99	58.57	61.09	59.27	61.86
4	-	-	-	-	-

Table 12: CoNLL 2002/2003 performances for the Future Languages (zero-shot setting, as in Figure 1c). At each step we report the average of the measure for the languages still not observed (step $> k$). The reported measure is the F1 (higher is better).

Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling

Chuhan Wu[†] **Fangzhao Wu**[‡] **Tao Qi**[†] **Yongfeng Huang**[†]
[†]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
[‡]Microsoft Research Asia, Beijing 100080, China
{wuchuhan15, wufangzhao, taoqi.qt}@gmail.com
yfhuang@tsinghua.edu.cn

Abstract

Transformer is important for text modeling. However, it has difficulty in handling long documents due to the quadratic complexity with input text length. In order to handle this problem, we propose a hierarchical interactive Transformer (Hi-Transformer) for efficient and effective long document modeling. Hi-Transformer models documents in a hierarchical way, i.e., first learns sentence representations and then learns document representations. It can effectively reduce the complexity and meanwhile capture global document context in the modeling of each sentence. More specifically, we first use a sentence Transformer to learn the representations of each sentence. Then we use a document Transformer to model the global document context from these sentence representations. Next, we use another sentence Transformer to enhance sentence modeling using the global document context. Finally, we use hierarchical pooling method to obtain document embedding. Extensive experiments on three benchmark datasets validate the efficiency and effectiveness of Hi-Transformer in long document modeling.

1 Introduction

Transformer (Vaswani et al., 2017) is an effective architecture for text modeling, and has been an essential component in many state-of-the-art NLP models like BERT (Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019; Wu et al., 2021). The standard Transformer needs to compute a dense self-attention matrix based on the interactions between each pair of tokens in text, where the computational complexity is proportional to the square of text length (Vaswani et al., 2017; Wu et al., 2020b). Thus, it is difficult for Transformer to model long documents efficiently (Child et al., 2019).

There are several methods to accelerate Transformer for long document modeling (Wu et al., 2019; Kitaev et al., 2019; Wang et al., 2020; Qiu et al., 2020). One direction is using Transformer in a hierarchical manner to reduce sequence length, e.g., first learn sentence representations and then learn document representations from sentence representations (Zhang et al., 2019; Yang et al., 2020). However, the modeling of sentences is agnostic to the global document context, which may be suboptimal because the local context within sentence is usually insufficient. Another direction is using a sparse self-attention matrix instead of a dense one. For example, Beltagy et al. (2020) proposed to combine local self-attention with a dilated sliding window and sparse global attention. Zaheer et al. (2020) proposed to incorporate a random sparse attention mechanism to model the interactions between a random set of tokens. However, these methods cannot fully model the global context of document (Tay et al., 2020).

In this paper, we propose a hierarchical interactive Transformer (*Hi-Transformer*)[1] for efficient and effective long document modeling, which models documents in a hierarchical way to effectively reduce the complexity and at the same time can capture the global document context for sentence modeling. In *Hi-Transformer*, we first use a sentence Transformer to learn the representation of each sentence within a document. Next, we use a document Transformer to model the global document context from these sentence representations. Then, we use another sentence Transformer to further improve the modeling of each sentence with the help of the global document context. Finally, we use hierarchical pooling method to obtain the document representation. Extensive experiments are conducted on three benchmark datasets. The results show that *Hi-Transformer* is both efficient and effective in long document modeling.

[1]https://github.com/wuch15/HiTransformer.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 848–853
August 1–6, 2021. ©2021 Association for Computational Linguistics

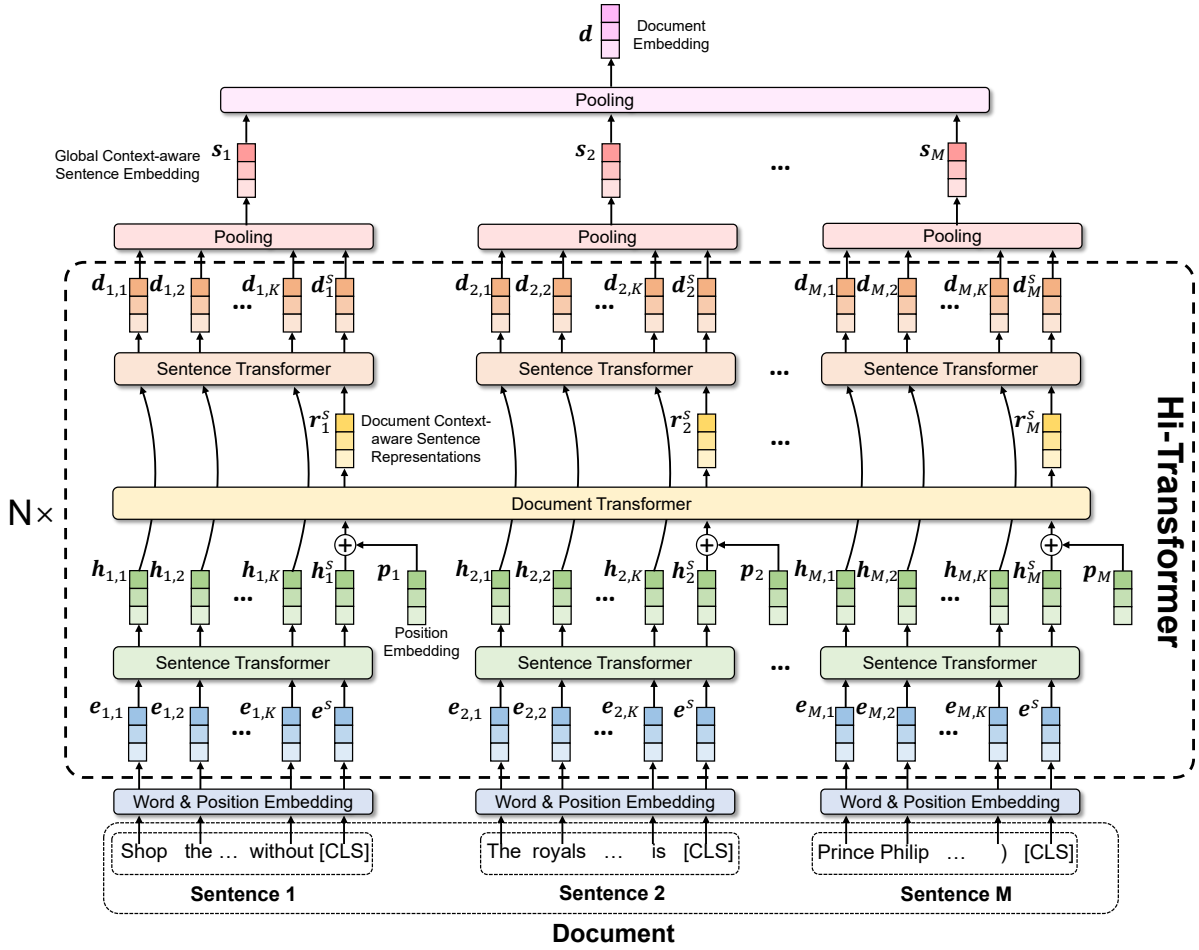


Figure 1: The architecture of *Hi-Transformer*.

2 Hi-Transformer

In this section, we introduce our hierarchical interactive Transformer (*Hi-Transformer*) approach for efficient and effective long document modeling. Its framework is shown in Fig. 1. It uses a hierarchical architecture that first models the contexts within a sentence, next models the document contexts by capturing the interactions between sentences, then employs the global document contexts to enhance sentence modeling, and finally uses hierarchical pooling techniques to obtain document embeddings. In this way, the input sequence length of each Transformer is much shorter than directly taking the word sequence in document as input, and the global contexts can be fully modeled. The details of *Hi-Transformer* are introduced as follows.

2.1 Model Architecture

Hi-Transformer mainly contains three modules, i.e., sentence context modeling, document context modeling and global document context-enhanced sentence modeling. The sentence-level context is first modeled by a sentence Transformer. Assume a document contains M sentences, and the words in the i-th sentence are denoted as $[w_{i,1}, w_{i,2}, ..., w_{i,K}]$ (K is the sentence length). We insert a "[CLS]" token (denoted as w^s) after the end of each sentence. This token is used to convey the contextual information within this sentence. The sequence of words in each sentence is first converted into a word embedding sequence via a word and position embedding layer. Denote the word embedding sequence for the i-th sentence as $[\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, ..., \mathbf{e}_{i,K}, \mathbf{e}^s]$. Since sentence length is usually short, we apply a sentence Transformer to each sentence to fully model the interactions between the words within this sentence. It takes the word embedding sequence as the input, and outputs the contextual representations of words, which are denoted as $[\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, ..., \mathbf{h}_{i,K}, \mathbf{h}_i^s]$. Specially, the representation $\mathbf{h}_i^s$ of the "[CLS]" token is regarded as the sentence representation.

Next, the document-level context is modeled by a document Transformer from the representations of the sentences within this document. Denote the

embedding sequence of sentences in this document as $[\mathbf{h}_1^s, \mathbf{h}_2^s, ..., \mathbf{h}_M^s]$. We add a sentence position embedding (denoted as $\mathbf{p}_i$ for the i-th sentence) to the sentence representations to capture sentence orders. We then apply a document Transformer to these sentence representations to capture the global context of document, and further learn document context-aware sentence representations, which are denoted as $[\mathbf{r}_1^s, \mathbf{r}_2^s, ..., \mathbf{r}_M^s]$.

Then, we use the document context-aware sentence representations to further improve the sentence context modeling by propagating the global document context to each sentence. Motivated by (Guo et al., 2019), we apply another sentence Transformer to the hidden word representations and the document-aware sentence representation for each sentence. It outputs a document context-aware word representation sequence for each sentence, which is denoted as $[\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, ..., \mathbf{d}_{i,K}, \mathbf{d}_i^s]$. In this way, the contextual representations of words can benefit from both local sentence context and global document context.

By stacking multiple layers of *Hi-Transformer*, the contexts within a document can be fully modeled. Finally, we use hierarchical pooling (Wu et al., 2020a) techniques to obtain the document embedding. We first aggregate the document context-aware word representations in each sentence into a global context-aware sentence embedding $\mathbf{s}_i$, and then aggregate the global context-aware embeddings of sentence within a document into a unified document embedding $\mathbf{d}$, which is further used for downstream tasks.

2.2 Efficiency Analysis

In this section, we provide some discussions on the computational complexity of *Hi-Transformer*. In sentence context modeling and document context propagation, the total computational complexity is $O(M \cdot K^2 \cdot d)$, where M is sentence number with a document, K is sentence length, and d is the hidden dimension. In document context modeling, the computational complexity is $O(M^2 \cdot d)$. Thus, the total computational cost is $O(M \cdot K^2 \cdot d + M^2 \cdot d)$.[2] Compared with the standard Transformer whose computational complexity is $O(M^2 \cdot K^2 \cdot d)$, *Hi-Transformer* is much more efficient.

[2]Note that *Hi-Transformer* can be combined with other existing techniques of efficient Transformer to further improve the efficiency for long document modeling.

3 Experiments

3.1 Datasets and Experimental Settings

Our experiments are conducted on three benchmark document modeling datasets. The first one is Amazon Electronics (He and McAuley, 2016) (denoted as Amazon), which is for product review rating prediction.[3] The second one is IMDB (Diao et al., 2014), a widely used dataset for movie review rating prediction.[4] The third one is the MIND dataset (Wu et al., 2020c), which is a large-scale dataset for news intelligence.[5] We use the content based news topic classification task on this dataset. The detailed dataset statistics are shown in Table 1.

In our experiments, we use the 300-dimensional pre-trained Glove (Pennington et al., 2014) embeddings for initializing word embeddings. We use two *Hi-Transformers* layers in our approach and two Transformer layers in other baseline methods.[6] We use attentive pooling (Yang et al., 2016) to implement the hierarchical pooling module. The hidden dimension is set to 256, i.e., 8 self-attention heads in total and the output dimension of each head is 32. Due to the limitation of GPU memory, the input sequence lengths of vanilla Transformer and its variants for long documents are 512 and 2048, respectively. The dropout (Srivastava et al., 2014) ratio is 0.2. The optimizer is Adam (Bengio and LeCun, 2015), and the learning rate is 1e-4. The maximum training epoch is 3. The models are implemented using the Keras library with Tensorflow backend. The GPU we used is GeForce GTX 1080 Ti with a memory of 11 GB. We use accuracy and macro-F scores as the performance metrics. We repeat each experiment 5 times and report both average results and standard deviations.

3.2 Performance Evaluation

We compare *Hi-Transformer* with several baselines, including: (1) *Transformer* (Vaswani et al., 2017), the vanilla Transformer architecture; (2) *Longformer* (Beltagy et al., 2020), a variant of Transformer with local and global attention for long documents; (3) *BigBird* (Zaheer et al., 2020), extending *Longformer* with random attention; (4) *HI-BERT* (Zhang et al., 2019), using Transformers

[3]https://jmcauley.ucsd.edu/data/amazon/

[4]https://github.com/nihalb/JMARS

[5]https://msnews.github.io/

[6]We also tried more Transformer layers for baseline methods but do not observe significant performance improvement in our experiments.

Dataset	#Train	#Val	#Test	Avg. #word	Avg. #sent	#Class
Amazon	40.0k	5.0k	5.0k	133.38	6.17	5
IMDB	108.5k	13.6k	13.6k	385.70	15.29	10
MIND	128.8k	16.1k	16.1k	505.46	25.14	18

Table 1: Statistics of datasets.

Methods	Amazon		IMDB		MIND	
	Accuracy	Macro-F	Accuracy	Macro-F	Accuracy	Macro-F
Transformer	65.23±0.38	42.23±0.37	51.98±0.48	42.76±0.49	80.96±0.22	59.97±0.24
Longformer	65.35±0.44	42.45±0.41	52.33±0.40	43.51±0.42	81.42±0.25	62.68±0.26
BigBird	66.05±0.48	42.89±0.46	52.87±0.51	43.79±0.50	81.81±0.29	63.44±0.31
HI-BERT	66.56±0.32	42.65±0.34	52.96±0.46	43.84±0.46	81.89±0.23	63.63±0.20
Hi-Transformer	67.24±0.35	43.69±0.32	53.78±0.49	44.54±0.47	82.51±0.25	64.22±0.22

Table 2: The results of different methods on different datasets.

Method	Complexity
Transformer	$O(M^2 \cdot K^2 \cdot d)$
Longformer	$O(T \cdot M \cdot K \cdot d)$
BigBird	$O(T \cdot M \cdot K \cdot d)$
HI-BERT	$O(M \cdot K^2 \cdot d + M^2 \cdot d)$
Hi-Transformer	$O(M \cdot K^2 \cdot d + M^2 \cdot d)$

Table 3: Complexity of different methods. K is sentence length, M is the number of sentences in a document, T is the number of positions for sparse attention, and d is the hidden dimension.

at both word and sentence levels. The results of these methods on the three datasets are shown in Table 2. We find that Transformers designed for long documents like *Hi-Transformer* and *BigBird* outperform the vanilla Transformer. This is because vanilla Transformer cannot handle long sequence due to the restriction of computation resources, and truncating the input sequence leads to the loss of much useful contextual information. In addition, *Hi-Transformer* and *HI-BERT* outperform *Longformer* and *BigBird*. This is because the sparse attention mechanism used in *Longformer* and *BigBird* cannot fully model the global contexts within a document. Besides, *Hi-Transformer* achieves the best performance, and the t-test results show the improvements over baselines are significant. This is because *Hi-Transformer* can incorporate global document contexts to enhance sentence modeling.

We also compare the computational complexity of these methods in Table 3. The complexity of *Hi-Transformer* is much less than the vanilla Transformer and is comparable with other Transformer variants designed for long documents. These results indicate the efficiency and effectiveness of *Hi-Transformer*.

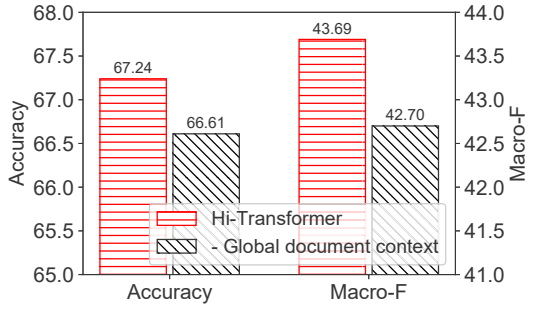
3.3 Model Effectiveness

Nest, we verify the effectiveness of the global document contexts for enhancing sentence modeling in *Hi-Transformer*. We compare *Hi-Transformer* and its variants without global document contexts in Fig. 2. We find the performance consistently declines when the global document contexts are not encoded into sentence representations. This is because the local contexts within a single sentence may be insufficient for accurate sentence modeling, and global contexts in the entire document can provide rich complementary information for sentence understanding. Thus, propagating the document contexts to enhance sentence modeling can improve long document modeling.

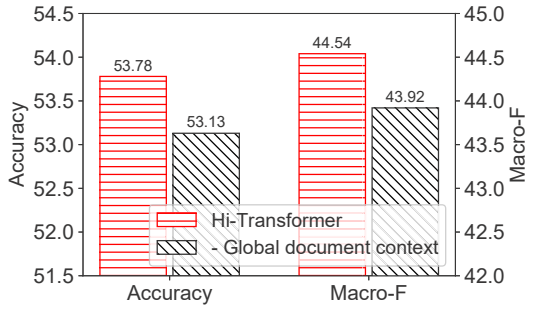
3.4 Influence of Text Length

Then, we study the influence of text length on the model performance and computational cost. Since the documents in the MIND dataset are longest, we conduct experiments on MIND to compare the model performance as well as the training time per layer of *Transformer* and *Hi-Transformer* under different input text length[7], and the results are shown in Fig. 3. We find the performance of both methods improves when longer text sequences are used. This is intuitive because more information can be incorporated when longer text is input to the model for document modeling. However, the computational cost of *Transformer* grows very fast,

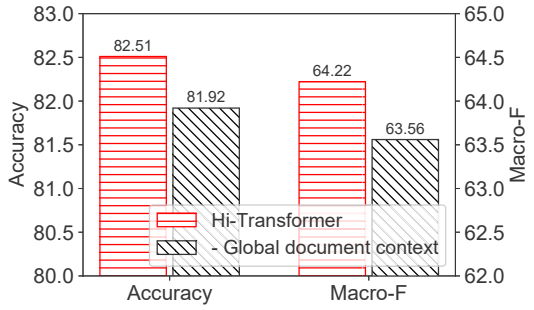
[7]The maximum length of *Transformer* is 512 due to GPU memory limitation.



(a) Amazon.

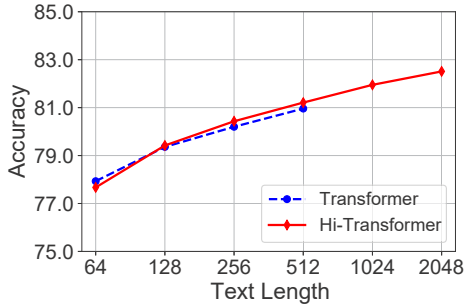


(b) IMDB.

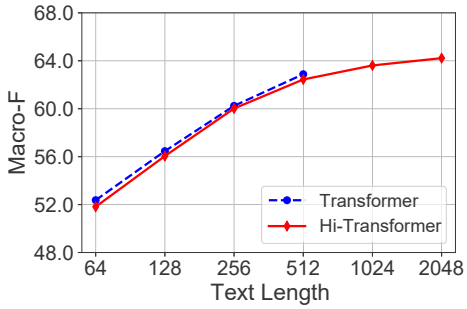


(c) MIND.

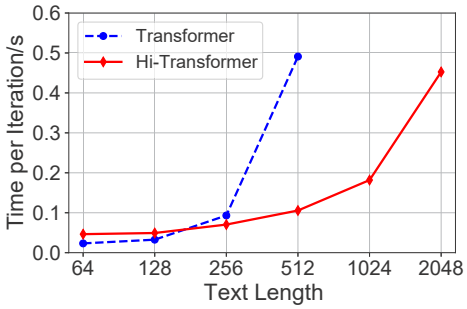
Figure 2: Effectiveness of global document context propagation in *Hi-Transformer* .



(a) Accuracy.



(b) Macro-F.



(c) Training time per layer.

Figure 3: Influence of input text length on performance and training time on the MIND dataset.

which limits its maximal input text length. Different from *Transformer*, *Hi-Transformer* is much more efficient and meanwhile can achieve better performance with longer sequence length. These results further verify the efficiency and effectiveness of *Hi-Transformer* in long document modeling.

4 Conclusion

In this paper, we propose a *Hi-Transformer* approach for both efficient and effective long document modeling. It incorporates a hierarchical architecture that first learns sentence representations and then learns document representations. It can effectively reduce the computational complexity and meanwhile be aware of the global document contexts in sentence modeling to help understand document content accurately. Extensive experiments on three benchmark datasets validate the efficiency and effectiveness of *Hi-Transformer* in long document modeling.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant numbers U1936216, U1936208, U1836204, and U1705261. We are grateful to Xing Xie, Shaoyu Zhou, Dan Shen, and Zhisong Wang for their insightful comments and suggestions on this work.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yoshua Bengio and Yann LeCun. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *KDD*, pages 193–202.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In *NAACL-HLT*, pages 1315–1325.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2019. Reformer: The efficient transformer. In *ICLR*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise self-attention for long document understanding. In *EMNLP: Findings*, pages 2555–2565.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. DA-transformer: Distance-aware transformer. In *NAACL-HLT*, pages 2059–2068.

Chuhan Wu, Fangzhao Wu, Tao Qi, Xiaohui Cui, and Yongfeng Huang. 2020a. Attentive pooling with learnable norms for text representation. In *ACL*, pages 2961–2970.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020b. Improving attention mechanism with query-value interaction. *arXiv preprint arXiv:2010.03766*.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020c. Mind: A large-scale dataset for news recommendation. In *ACL*, pages 3597–3606.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. 2019. Lite transformer with long-short range attention. In *ICLR*.

Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In *CIKM*, pages 1725–1734.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5753–5763.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL*, pages 5059–5069.

Robust Transfer Learning with Pretrained Language Models through Adapters

Wenjuan Han[1*†], **Bo Pang**[2*], **Yingnian Wu**[2]
[1] Beijing Institute for General Artificial Intelligence, Beijing, China
[2] Department of Statistics, University of California, Los Angeles
hanwenjuan@bigai.ai
{bopang, ywu}@ucla.edu

Abstract

Transfer learning with large pretrained transformer-based language models like BERT has become a dominating approach for most NLP tasks. Simply fine-tuning those large language models on downstream tasks or combining it with task-specific pretraining is often not robust. In particular, the performance considerably varies as the random seed changes or the number of pretraining and/or fine-tuning iterations varies, and the fine-tuned model is vulnerable to adversarial attack. We propose a simple yet effective adapter-based approach to mitigate these issues. Specifically, we insert small bottleneck layers (i.e., adapter) within each layer of a pretrained model, then fix the pretrained layers and train the adapter layers on the downstream task data, with (1) task-specific unsupervised pretraining and then (2) task-specific supervised training (e.g., classification, sequence labeling). Our experiments demonstrate that such a training scheme leads to improved stability and adversarial robustness in transfer learning to various downstream tasks. [1]

1 Introduction

Pretrained transformer-based language models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) have demonstrated impressive performance on various NLP tasks such as sentiment analysis, question answering, text generation, just to name a few. Their successes are achieved through sequential transfer learning (Ruder, 2019): pretrain a language model on large-scale unlabeled data and then fine-tune it on downstream tasks with labeled data. The most commonly used fine-tuning approach is to optimize all parameters of the pretrained model with regard to the downstream-task-specific loss. This training scheme is widely adopted due to its simplicity and flexibility (Phang et al., 2018; Peters et al., 2019; Lan et al., 2019; Raffel et al., 2020; Clark et al., 2020; Nijkamp et al., 2021; Lewis et al., 2020).

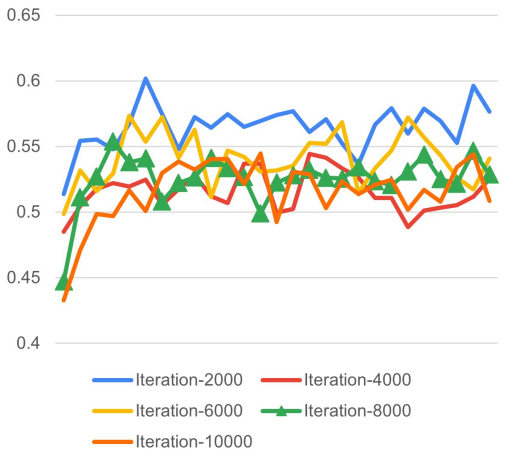


Figure 1: Learning curves of fine-tuning with the task-specific pretraining iterations varied. The curve with triangles represents the model that has converged in the 8000-th pretraining iteration.

Despite the success of the standard sequential transfer learning approach, recent works (Gururangan et al., 2020; Lee et al., 2020; Nguyen et al., 2020) have explored domain-specific or task-specific unsupervised pretraining, that is, masked language model training on the downstream task data before the final supervised fine-tuning on it. And they demonstrated benefits of task-specific pretraining on transfer learning performance. However, both standard sequential transfer learning and that with task-specific pretraining are unstable in the sense that downstream task performance is subject to considerable fluctuation while the random seed is changed or the number of pretraining and/or fine-tuning iterations is varied even after the training has converged (see Section 2 and Section 3

[*] Equal contributions.
[†] Corresponding author.
[1] https://github.com/WinnieHAN/Adapter-Robustness.git

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 854–861
August 1–6, 2021. ©2021 Association for Computational Linguistics

		WNLI	RTE	MRPC	STS-B	CoLA	SST-2	QNLI	QQP	MNLI
Metrics		*Acc.*	*Acc.*	*F1/Acc.*	*P/S corr.*	*M corr.*	*Acc.*	*Acc.*	*Acc./F1*	*M acc.*
WO.		56.34	65.7	88.85/84.07	88.64/88.48	56.53	92.32	90.66	90.71/87.49	84.10
W.	*F.*	45.07	61.73	89.47/85.29	83.95/83.70	49.23	91.97	87.46	88.40/84.31	81.08
	TSP.+F.	56.34	68.59	89.76/86.37	89.24/88.87	64.87	92.78	91.12	90.92/87.88	84.14

Table 1: Performance on the development dataset of GLUE. Results of W.(F.) are reported in Adapter-Hub. We report results of WO. using the implementation from Wolf et al. (2020). Acc.: Accuracy. M acc.: Mismatched Acc. P/S acc.: Person/Spearman corr. M corr.: Matthew's corr. TSP.: Task-Specific Pretrain. F.: Finetune. WO.: Without adapter. W.: With adapter.

for details). For instance, as observed in Fig. 1, as the number of task-specific pretraining iteration varies, CoLA's performance is severely unstable in fine-tuning. Besides instability, we also observe that task-specific pretraining is vulnerable to adversarial attack. Last but not least, task-specific pretraining and/or fine-tuning on the entire model is highly parameter-inefficient given the large size of these models (e.g., the smallest BERT has 110 million parameters).

In this work, we propose a simple yet effective adapter-based approach to mitigate these issues. Adapters are some small bottleneck layers inserted within each layer of a pretrained model (Houlsby et al., 2019; Pfeiffer et al., 2020a,b). The adapter layers are much smaller than the pretrained model in terms of the number of parameters. For instance, the adapter used in (Houlsby et al., 2019) only adds 3.6% parameters per task. In our approach, we adapt the pretrained model to a downstream task through 1) task-specific pretraining and 2) task-specific supervised training (namely, fine-tuning) on the downstream task (e.g., classification, sequence labeling) by only optimizing the adapters and keeping all other layers fixed. Our approach is parameter-efficient given that only a small number of parameters are learned in the adaptation.

The adapted model learned through our approach can be viewed as a residual form of the original pretrained model. Suppose x is an input sequence and h_{original} is the features of x computed by the original model. Then the feature computed by the adapted model is,

$$h_{\text{adapted}} = h_{\text{original}} + f_{\text{adapter}}(x), \qquad (1)$$

where $f_{\text{adapter}}(x)$ is the residual feature in addition to h_{original} and f_{adapter} is the adapter learned in the adaptation process. h_{original} extracts general features that are shared across tasks, while f_{adapter} is learned to extract task-specific features. In prior work (Houlsby et al., 2019; Pfeiffer et al., 2020b), f_{adapter} is learned with task-specific supervised learning objective, distinctive from the unsupervised pretraining objective, and might not be compatible with h_{original}, as evidenced in our experiments. In our approach, f_{adapter} is first trained with the same pretraining objective[2] on the task-specific data before being adapted with the supervised training objective, encouraging the compatibility between h_{original} and f_{adapter}, which is shown to improve the downstream task performance in our experiments (see Table 1).

Some prior works have examined the potential causes of the instability of pretrained language models in transfer learning. Lee et al. (2019) proposed that catastrophic forgetting in sequential transfer learning underlined the instability, while Mosbach et al. (2020) proposed that gradient vanishing in fine-tuning caused it. Pinpointing the cause of transfer learning instability is not the focus of the current work, but our proposed method seems to able to enhance transfer learning on both aspects.

The standard sequential transfer learning or that with task-specific pretraining updates all model parameters in fine-tuning. In contrast, our approach keeps the pretrained parameters unchanged and only updates the parameters in the adapter layers, which are a small amount compared to the pretrained parameters. Therefore, our approach naturally alleviates catastrophic forgetting considering the close distance between the original pretrained model and the adapted model. On the other hand, we do not observe gradient vanishing with our transfer learning scheme (see Section 2 for more details). This might be because optimizing over a much smaller parameter space in our approach, compared to the standard sequential transfer learning scheme where all parameters are trained, renders the op-

[2]In this work, we conduct experiments with the most widely used pretraining objective, masked language modeling. The same training scheme can be extended to other pretraining objectives.

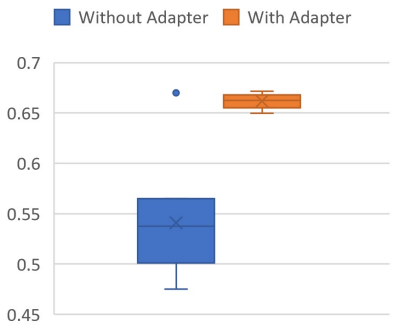


Figure 2: Distribution of dev scores on RTE from 10 random seed restarts when finetuning (1) BERT (Devlin et al., 2019) and (2) BERT with the adapter architecture.

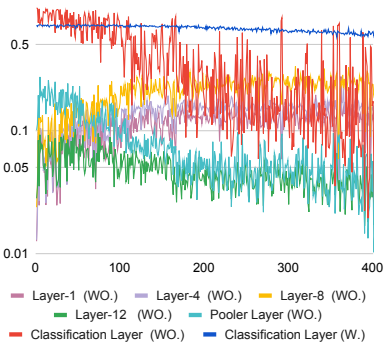


Figure 3: Gradient norms (on log scale) of intermediate layer and classification layer on RTE for with/without-adapter fine-tuning run. WO.: Without adapter. W.: With adapter.

timization easier. We leave it to future work for further theoretical analysis.

In addition to its improved stability, the proposed transfer learning scheme is also likely to be more robust to adversarial attack. Given that it updates the entire model, the standard transfer learning approach might suffer from overfitting to the downstream task, and thus a small perturbation in the input might result in consequential change in the model prediction. In turn, it might be susceptible to adversarial attack. Our approach only updates a much smaller portion of parameters, and hence might be more robust to these attacks, which is confirmed in our empirical analysis (see Section 4).

Contributions. In summary our work has the following contributions. (1) We propose a simple and parameter-efficient approach for transfer learning. (2) We demonstrate that our approach improves the stability of the adaptation training and adversarial robustness in downstream tasks. (3) We show the improved performance of our approach over strong baselines. Our source code is publicly available at https://github.com/WinnieHAN/Adapter-Robustness.git.

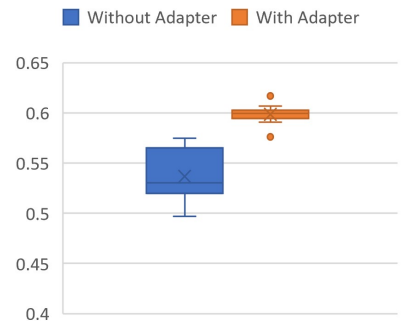


Figure 4: Box plots showing the TSP. stability of BERT with/without adapter on CoLA.

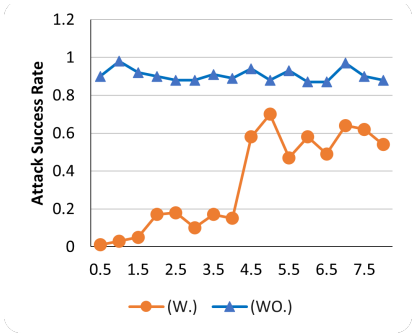


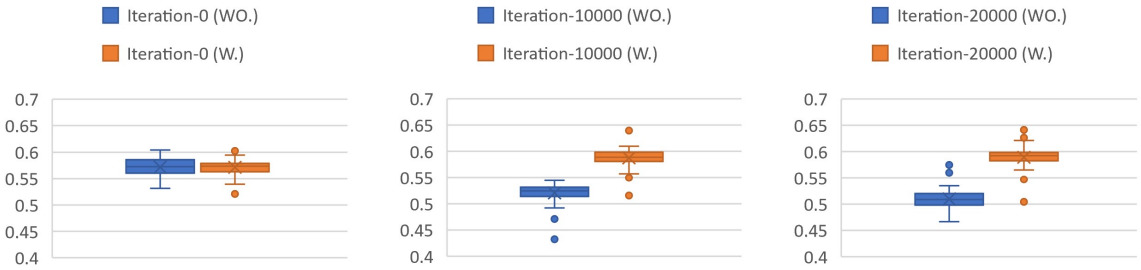
Figure 5: Attack success rate of BERT with/without adapter during task-specific pretraining. WO.: Without adapter. W.: With adapter.

2 Instability to Different Random Seeds

We first evaluate the training instability with respect to multiple random seeds: fine-tuning the model multiple times in the same setting, varying only the random seed. We conduct the experiments on RTE (Wang et al., 2018) when fine-tuning 1) BERT-base-uncased (Devlin et al., 2019) and 2) BERT-base-uncased with the adapter (Houlsby et al., 2019) [3]. As shown in Figure 2, the model without adapter leads to a large standard deviation on the fine-tuning accuracy, while the one with adapter results in a much smaller variance on the task performance.

Gradient Vanishing Mosbach et al. (2020) argues that the fine-tuning instability can be explained by optimization difficulty and gradient vanishing. In order to inspect if the adapter-based approach suffers from this optimization problem, we plot the L_2 gradient norm with respect to different layers of BERT, pooler layer and classification layer, for fine-tuning with or without adapter in

[3]For all the experiments, we use the implementation of Pfeiffer et al. (2020b): https://github.com/Adapter-Hub/adapter-transformers.git.



(a) Pretraining Iteration 0 (b) Pretraining Iteration 10000 (c) Pretraining Iteration 20000

Figure 9: Box plots showing the fine-tuning stability of BERT with/without adapter for different TSP. iterations on CoLA. WO.: Without adapter. W.: With adapter.

Figure 3.

In traditional fine-tuning (without adapter), we see vanishing gradients for not only the top layers but also the pooler layer and classification layer. This is in large contrast to the with-adapter fine-tuning. The gradient norm in the with-adapter fine-tuning does not decrease significantly in the training process. These results imply that the adaptation with adapter does not exhibit gradient vanishing and presents a less difficult optimization problem, which in turn might explain the improved stability of our approach.

3 Instability to Pretraining and Fine-tuning Iterations

Fine-tuning with all parameters also exhibits another instability issue. In particular, fine-tuning a model multiple times on the pretrained language model, varying the task-specific pretraining iterations and fine-tuning iterations, leads to a large standard deviation in downstream task performance. As observed in Figure 1, CoLA's performance when varying the task-specific pretraining iterations is severely unstable during pretraining iterations and fine-tuning iterations. The model has converged at the pretraining iteration of 8000. However, fine-tuning based on this model does not obtain the best performance.

Pretraining Iterations. Figure 4 displays the performance on CoLA of 10 fine-tuning runs with and without the adapter. For each run, we vary only the number of pretraining iterations from 2000 to 20000 with an interval of 2000 and fix the fine-tuning epochs to 10. We clearly observe that most runs for BERT with adapter outperforms the one without adapter. Moreover, the adapter makes pretraining BERT significantly more stable than the standard approach (without adapter).

Fine-tuning Iterations. We then study the stability with regard to the number of fine-tuning iterations. We show box plots for BERT using various pretraining iterations and fine-tuning iterations, with and without adapter in Figure 9. The three sub-figures represent the early, mid, and late stages of pretraining, corresponding to the 0-th, 10000-th, and 20000-th iteration respectively. The 0-th iteration represents the original model without task-specific pretraining. The model suffers from underfitting in the 0-th iteration and overfitting in the 20000-th iteration.

In Figure 9 (a), we plot the distributions of the development scores from 100 runs when fine-tuning BERT with various fine-tuning epochs ranging from 1 to 100. In the early stage, the average development score of the model with the adapter is a little lower than the baseline model while the stability is better. After several epochs of pretraining, the adapter gradually shows improved performance in terms of the mean, minimum and maximum as demonstrated in Figure 9 (b). In the end of the pretrainig, there exists an over-fitting problem for the traditional BERT models. Pretraining transfers the model to a specific domain and fails to maintain the original knowledge. In contrast, the performance with the adapter still grows as training continues and consistently benefit from pretraining. Besides, we observe that the adapter leads to a small variance in the fine-tuning performance, especially in the late stage. Additional plots and learning curves can be found in the Appendix.

4 Adversarial Robustness

While successfully applied to many domains, the predictions of Transformers (Vaswani et al., 2017)

become unreliable in the presence of small adversarial perturbations to the input (Sun et al., 2020; Li et al., 2020). Therefore, the adversarial attacker has become an important tool (Moosavi-Dezfooli et al., 2016) to verify the robustness of models. The robustness is usually evaluated from attack effectiveness (i.e., attack success rate). We use a SOTA adversarial attack approach to assess the robustness: PWWS attacker (Ren et al., 2019). [4]. Figure 5 shows the attack success rate of BERT with/without adapter during task-specific pretraining on SST-2. The x-axis is the number of epochs for task-specific pretraining. It can be observed that the model with the adapter has better adversarial robustness.

5 Conclusion

We propose a simple yet effective transfer learning scheme for large-scale pretrained language model. We insert small bottleneck layers (i.e., adapter) within each block of the pretrained model and then optimize the adapter layers in task-specific unsupervised pretraining and supervised training (i.e., fine-tuning) while fixing the pretrained layers. Extensive experiments demonstrate that our approach leads to improved stability with respect to different random seeds and different number of iterations in task-specific pretraining and fine-tuning, enhanced adversarial robustness, and better transfer learning task performance. We therefore consider the proposed training scheme as a robust and parameter-efficient transfer learning approach.

Acknowledgments

Y. W. is partially supported by NSF DMS 2015577.

References

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2019. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. *Advances in Neural Information Processing Systems*, 33.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

[4] We use the implementation in OpenAttack toolkit https://github.com/thunlp/OpenAttack.git. It generates adversarial examples and evaluation the adversarial robustness of the victime model using thee adversarial examples. We use the default settings including all the hyper-parameter values.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Erik Nijkamp, Bo Pang, Ying Nian Wu, and Caiming Xiong. 2021. SCRIPT: Self-critic PreTraining of transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5196–5202, Online. Association for Computational Linguistics.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097.

Sebastian Ruder. 2019. *Neural transfer learning for natural language processing*. Ph.D. thesis, NUI Galway.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Hyper-Parameters Setting

We conduct the experiments on the task of GLUE tasks (Wang et al., 2018) when fine-tuning 1) BERT-base-uncased (Devlin et al., 2019) and 2) BERT-base-uncased with the adapter architecture (Houlsby et al., 2019). For all the experiments, we use the implementation from `https://github.com/Adapter-Hub/adapter-transformers.git`. For the model with adapter, we follows the setup from Mosbach et al. (2020). For all experiments, we use the default hyper-parameters except for the number of epochs. Please refer to the provided link.

The main hyper-parameters are listed in Table 2 and Table 3.

Max Sequence Length	256
Batch Size	32
Learning rate	1e-4
Number of Epochs	20

Table 2: Hyper-parameters for BERT with Adapter.

Max Sequence Length	128
Batch Size	32
Learning rate	2e-5
Number of Epochs	10

Table 3: Hyper-parameters for BERT without Adapter.

B Instability to Pretraining and Fine-tuning Iterations

We provide box plots for BERT using various pre-training iterations and fine-tuning iterations, with and without adapter on CoLA in Figure 10. The corresponding learning curves are in Figure 13.

C Instability for Large Dataset

In contrast to relatively large datasets, smaller data is more suitable and convincing as an example to analyze stability. Small dataset is easier to encounter over-fitting problems and often not stable (Devlin et al., 2019). We use MNLI to evaluate the training instability in terms of 5 random seeds with the same setup in Figure 2. The interquartile range of BERT with adapter on the distribution of dev scores is smaller than BERT without adapter. It shows that the model without adapter consistently leads to the instability issue on the fine-tuning accuracy, while the adapter architecture brings less benefit with larger dataset.

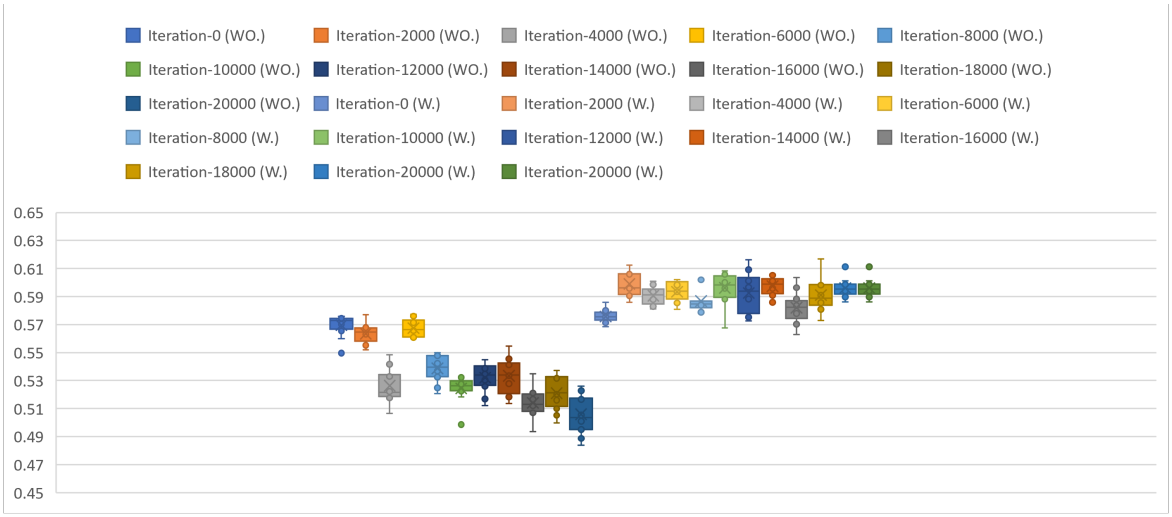
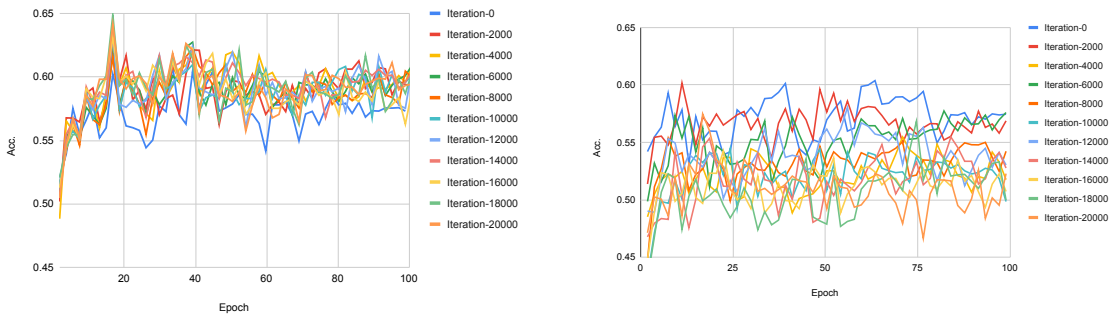


Figure 10: Box plots showing the fine-tuning stability of BERT with/without adapter for different pretraining iteration from 0 to 20000.



(a) BERT with adapter.

(b) BERT without adapter.

Figure 13: Learning curves of fine-tuning when varying the pretraining iterations.

Embracing Ambiguity: Shifting the Training Target of NLI Models

Johannes Mario Meissner[†], **Napat Thumwanit**[†], **Saku Sugawara**[‡], **Akiko Aizawa**[†‡]
[†]The University of Tokyo, [‡]National Institute of Informatics
{meissner,thumwanit-n,saku,aizawa}@nii.ac.jp

Abstract

Natural Language Inference (NLI) datasets contain examples with highly ambiguous labels. While many research works do not pay much attention to this fact, several recent efforts have been made to acknowledge and embrace the existence of ambiguity, such as UNLI and ChaosNLI. In this paper, we explore the option of training directly on the estimated label distribution of the annotators in the NLI task, using a learning loss based on this ambiguity distribution instead of the gold-labels. We prepare AmbiNLI, a trial dataset obtained from readily available sources, and show it is possible to reduce ChaosNLI divergence scores when finetuning on this data, a promising first step towards learning how to capture linguistic ambiguity. Additionally, we show that training on the same amount of data but targeting the ambiguity distribution instead of gold-labels can result in models that achieve higher performance and learn better representations for downstream tasks.

1 Introduction

Ambiguity is intrinsic to natural language, and creating datasets free of this property is a hard if not impossible task. Previously, it was common to disregard it as noise or as a sign of poor quality data. More recent research, however, has drawn our attention towards the inevitability of ambiguity, and the necessity to take it into consideration when working on natural language understanding tasks (Pavlick and Kwiatkowski, 2019; Chen et al., 2020; Nie et al., 2020; Swayamdipta et al., 2020). This ambiguity stems from the lack of proper context or differences in background knowledge between annotators, and leads to a large number of examples where the correctness of labels can be debated.

ChaosNLI (Nie et al., 2020) is a dataset created by manually annotating a subset of the SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and αNLI (Bhagavatula et al., 2020) datasets. Each of the total 4,645 samples received 100 annotations. Through this data, they were able to generate a probability distribution over the labels for these samples, which they call the human agreement distribution, with the goal of using it to evaluate the ability of current state-of-the-art models to capture ambiguity. The divergence scores between the model's predicted probability distribution and the true target distribution is computed and compared against random and human baselines. They showed that models trained using gold-labels have very poor performance on the task of capturing the human agreement distribution.

Although this is a promising first step, it remains unclear how to train models with a better understanding of ambiguity, and what tangible benefits we can obtain when actually doing so. In this work, we study the possibility of shifting the training target of models from gold-labels to the ambiguity distribution, a simple and intuitive yet until now unexplored approach in this domain. We hypothesize that when we finetune a model in this way, we can achieve lower divergence scores in the ChaosNLI benchmark. Further, we believe that it should also bring accuracy improvements in NLI and other downstream tasks. The intuition behind our performance expectations is that an ambiguity distribution offers a more informative and less misleading view on the answer to the task, which allows models to learn more from the same data.

We prepare a trial dataset with ambiguity distributions obtained from available SNLI and MNLI data, and run experiments to confirm our hypotheses. We refer to it as AmbiNLI, but we do not encourage its use in further work. Instead, we encourage the community to follow this direction by performing further data collection in this area.

Our main contributions are showing that 1) models trained on ambiguity can more closely capture

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 862–869
August 1–6, 2021. ©2021 Association for Computational Linguistics

Dataset	Split	Used By	#Samples	#Labels
SNLI	Train	UNLI	55,517	1r
	Dev.	UNLI	3,040	1r
		ChaosNLI	1,514	100
		Original	9,842	5
	Test	UNLI	3,040	1r
		Original	9,824	5
MNLI	Dev. M.	ChaosNLI	1,599	100
		Original	9,815	5
	Dev. Mism.	Original	9,832	5

Table 1: Data with enough information to generate a probability distribution over the labels. The marker "1r" denotes the fact that there is only one data-point available, but it is a regression label in the [0,1] range, so it can be converted.

Data	ChaosSNLI		ChaosMNLI	
Metric	JSD↓	Acc.↑	JSD↓	Acc.↑
S/MNLI Baseline	0.2379	0.7497	0.3349	0.5566
+ AmbiSM Gold	0.2307	0.7497	0.3017	0.5660
+ AmbiSM	**0.1893**	**0.7550**	0.2619	0.5810
+ AmbiU Gold	0.3118	0.5878	0.3183	0.5260
+ AmbiU	0.2834	0.5964	0.2843	0.5178
+ AmbiU Filt.	0.2302	0.6790	**0.2231**	0.5779
+ AmbiSMU Gold	0.2936	0.6162	0.3540	0.5822
+ AmbiSMU	0.2554	0.6420	0.2575	0.5766
+ AmbiSMU Filt.	0.2155	0.7107	0.2748	**0.5835**

Table 2: Main results of our finetuning experiments on AmbiNLI. *Gold* means that gold-labels, and not ambiguity distribution, was used for training. *Filt.* indicates that extreme examples in UNLI have been filtered out.

the true human distribution, 2) they are able to attain higher accuracy under otherwise equal conditions, and 3) they learn better representations for downstream tasks. We release the code used for these experiments.[1]

2 AmbiNLI

2.1 Available Data

Data containing enough information to reveal ambiguity is relatively scarce. To construct AmbiNLI we generated the label distributions from several sources. Table 1 details the available data that we have taken into consideration.

SNLI / MNLI. Both SNLI and MNLI provide labels assigned by 5 annotators on some subsets of the data (marked "Original" in Table 1). Examples where no human agreement could be reached (no majority) were given a special label (-1) and are commonly filtered out. Although the precision given by 5 labels is much lower than that of the 100 annotations provided in ChaosNLI, we believe that even a rough and inaccurate ambiguity representation is more beneficial than gold-labels only.

UNLI. UNLI (Chen et al., 2020) presents a subset of SNLI as a regression task, where each example is annotated with a real value in the range [0,1]. Values close to 0 indicate contradiction, and values close to 1 represent entailment. Each entry has one label only, but since it real-valued, it is also possible to extract a distribution from it. Even though it seems to be a less suitable data source, we do intend to investigate its effectiveness for our purposes.

ChaosNLI. ChaosNLI provides annotations from 100 humans for 3,113 examples in the development sets of SNLI and MNLI. We will call these subsets ChaosSNLI and ChaosMNLI. In order to allow for comparison with the original paper, we use them for testing only.

2.2 Creating AmbiNLI

Original SNLI and MNLI data with 5 annotations can be converted to an ambiguity distribution by simply counting the number of annotations for each label and then scaling it down into probabilities. We make sure to avoid overlap between ChaosNLI and "Original" data by removing the samples used in ChaosNLI from the data we will include in AmbiNLI. In the case of UNLI, we have taken only the 55,517 samples from the training set, so there is no overlap with ChaosNLI. We apply a simple linear approach to convert the UNLI regression value p into a probability distribution z_{NLI}, as described in the following composed function (its plot can be found in the Appendix A):

$$z_{\text{NLI}} = \begin{cases} (0, 2p, 1-2p) & p < 0.5 \\ (2p-1, 2-2p, 0) & p \geq 0.5. \end{cases}$$

The resulting AmbiNLI dataset has 18,152 SNLI examples, 18,048 MNLI examples, and 55,517 UNLI examples, for a total of 91,717 premise-hypothesis pairs with an ambiguity distribution as the target label.

[1] https://github.com/mariomeissner/AmbiNLI

3 Experiments

In our experiments, we use BERT-base (Devlin et al., 2019) with pre-trained weights and a softmax classification head. We use a batch size of 128 and learning rate of 1e-5.

Learning to capture question ambiguity. In our main experiment, we aim to judge whether is is possible to learn how to capture the human agreement distribution. We first obtain a base model in the same manner as Nie et al. (2020), by pre-training it for 3 epochs on the gold-labels of the SNLI and MNLI training sets. We observed that this pre-training step is necessary to provide the model with a general understanding of the NLI task to compensate for the low amount of ambiguity data available. We then finetune the model on our AmbiNLI dataset, setting the training objective to be the minimization of the cross-entropy between the output probability distribution and the target ambiguity distribution. For evaluation, we compute the ChaosNLI divergence scores, measured using the Jensen-Shannon Divergence (JSD), as was done in their original experiments. Furthermore, we explore what effect our ambiguity learning has on accuracy by comparing models trained on exactly the same data but with gold-label training versus ambiguous training. In order to achieve this, we prepare a version of AmbiNLI where we replace the ambiguity distributions with gold-labels. Since the two models have seen the exact same data, performance differences can be directly attributed to the process of capturing ambiguity. We report accuracy on ChaosNLI using their re-computed gold-labels.

Further accuracy analysis. To reinforce our hypothesis that accuracy improvements can be gained by leveraging the extra knowledge that models capture with ambiguity, we run an additional experiment on the ChaosMNLI dataset. We split it into three folds, and perform three-fold cross validation by training the model on two folds and evaluating on the third. Again, we start with our baseline model and compare the gold-label approach against ours.

Performance in different entropy ranges. We also study the model performance in different entropy ranges of the ChaosMNLI set. We bin the evaluation samples based on their entropy value into three equally sized ranges, and compare the model performance on each. This experiment analyzes if the model is able to perform well in both unambiguous and highly ambiguous settings.

Folds	AmbiSM Gold	AmbiSM
0	0.4371	**0.4409**
1	**0.5760**	0.5591
2	0.4897	**0.5629**
Average	0.5009	**0.5210**

Table 3: Model accuracy when performing three-fold cross validation of a BERT base model on ChaosMNLI.

Entropy Range	JSD	Accuracy
Full Range	0.2619	0.5810
[0.08 - 0.58]	0.2613	**0.6706**
[0.58 - 1.08]	**0.2472**	0.6262
[1.08 - 1.58]	0.2693	0.5087

Table 4: Entropy range performance comparison of the AmbiSM model.

Transfer learning. In this last experiment, we aim to compare the usefulness of the representations that the BERT encoder is able to learn when training on ambiguity distributions as opposed to gold-labels. We use UNLI and IMBD movie reviews (Maas et al., 2011) as the two downstream tasks for evaluation. As we want to focus on the representations learned during the ambiguity training phase, during the downstream task finetuning we freeze the BERT layers and update only the new classification head. We try with 1-layer and 2-layer heads using the ELU (Clevert et al., 2016) activation function and a hidden size of 128. We use the original train, development and test splits for UNLI, and an 80/10/10% split for IMDB movie reviews. We track development set loss and stop after two epochs without improvement. Each experiment is ran for 5 trials with different seeds and the mean and standard deviation are reported for each metric.

4 Results and Discussion

Training on the ambiguity distribution can reduce divergence scores. Table 2 details the results of our main experiment. Accuracy and JSD are provided for both the SNLI and MNLI sections in ChaosNLI. Due to differences in hyperparameters or random seeds, we were not able to exactly reproduce the base model provided in Nie et al. (2020), but achieve similar results. We follow with models further finetuned on different config-

urations of our AmbiNLI dataset. AmbiSM refers to the data originating from the original 5 label distribution only, while AmbiU refers to the data we obtained from UNLI. AmbiSMU thus refers to the full dataset. For each combination, we also trained a model on gold-labels (marked as "Gold" in the table) for comparison. With the exception of ChaosSNLI when including UNLI data, every experiment has yielded a mentionable divergence score improvement. The AmbiSM model shows a 20.5% and 21.7% JSD decrease in ChaosSNLI and ChaosMNLI respectively. This means that we can learn to capture the human agreement distribution when we use it as a training target.

UNLI's skewed distribution worsens scores. When looking at the AmbiU and AmbiSMU results in Table 2, it becomes apparent that UNLI data is not always beneficial. Specifically, it seems to worsen scores in all metrics except for ChaosMNLI accuracy. The distribution of labels in UNLI is drastically different from that of the remaining data, and we believe that when a model is finetuned on it, this distribution shift has a negative influence. We have found a very large number of samples with labels very close to 0 or 1, which translate into very extreme non-ambiguous distributions when converted. To confirm this, we filtered out all UNLI samples that had a probability label $p < 0.05$ or $p > 0.97$, and ran the "Filtered" experiments. Indeed, in AmbiU, this naive filtering process yields about 20% lower JSD scores and about 5% higher accuracy. We conclude that UNLI data, under the current conversion approach, is somewhat problematic.

Training on the ambiguity distribution can yield accuracy improvements. We have found that, for the case of AmbiSM, a model trained to target the ambiguity distribution achieves higher accuracy. This means that more precise knowledge can be acquired when learning the true underlying ambiguity of questions instead of the sometimes misleading gold-label. When using UNLI data (AmbiU and AmbiSMU) however, the results are mixed, as discussed above. Thus, to further strengthen our argument on the benefit of ambiguity data, we refer to the supplementary experiment results in Table 3, where we obtain a 2.1% accuracy improvement when performing three-fold cross-validation on the ChaosMNLI dataset. When performing a qualitative analysis on the predictions of the AmbiSM and AmbiSM Gold models, we found that the former has a stronger tendency towards neutrality, both in the number of correctly predicted neutral labels and in the average neutrality score given. However, it also resulted in some examples now being incorrectly labeled as neutral. It seems to be the case that the neutral label is the main source of ambiguity. Most ambiguous questions have a considerable amount of neutral probability, which likely produces the shift. For more details, including label counts for correct predictions as well as some prediction examples, refer to Appendix B.

Divergence scores are stable. Through the entropy range comparison of Table 4 we learn that divergence scores remain similar across different entropy subsets, showing that the model is capable of recognizing which questions are ambiguous, and appropriately adjusting the entropy level of its output. Accuracy dramatically decreases in high entropy ranges, but this goes along with our intuition, since both human annotators and the models will have doubts regarding the correct answer to the question, which leads to mismatches between the model prediction and the assigned label.

Ambiguity models learn better representations for transfer learning. Lastly, in Table 5, we observe a consistent performance improvement in transfer learning to different tasks. From the results we can infer that, by targeting the ambiguity distribution, the model can capture better linguistic representations than by targeting gold-labels. We believe that a similar trend should be visible in other tasks as well, and that the margins of improvement should increase with more ambiguity data to train on.

Is ambiguity training worth the extra labeling cost? One argument against this method is the apparent extra labeling cost required. Indeed, when comparing the gold-label and ambiguity approaches at equal number of total labels, the gold-label approach would likely attain higher performance due to the difference in number of samples. However, we argue that collecting multiple labels has several benefits other than ambiguity distribution generation. Most importantly, they help avoid mis-labelings and raise the overall quality of the dataset. In many occasions, multiple labels are already being collected for these reasons, but occasionally not released (for example, Bowman et al. (2015) didn't release the multiple labels they col-

Model	UNLI Pearson↑	UNLI MSE↓	IMDB CE Loss↓	IMDB Acc.↑
1 Layer				
AmbiSM G	.6331(0.9)	.0758(0.5)	.4727(1.7)	.7758(6.4)
AmbiSM	**.6354(1.0)**	**.0754(0.4)**	**.4701(1.5)**	**.7783(6.1)**
2 Layers				
AmbiSM G	.6266(5.9)	.0765(1.0)	.4431(0.8)	.7906(4.3)
AmbiSM	**.6290(4.1)**	**.0762(0.7)**	**.4392(1.2)**	**.7939(3.3)**

Table 5: Transfer learning comparison on UNLI and IMDB movie reviews (*std* is $\times 10^{-4}$). For UNLI we measure the Pearson correlation and mean squared error (MSE), following Chen et al. (2020). For IMDB, we measure the accuracy and cross-entropy (CE) loss on the test set. *G* means Gold.

lected for 10% of the training data). They can also be used in other methods such as item response theory (Lalor et al., 2016). Furthermore, this paper's main intention is not to encourage multi-label collection at the cost of sample quantity, but rather to show the benefits of exploiting the ambiguity distribution if it is available.

5 Conclusion

We hypothesized that the intrinsic ambiguity present in natural language datasets can be exploited instead of treating it like noise. We used existing data to generate ambiguity distributions for subsets of SNLI, MNLI, and UNLI, and trained new models that are capable of more accurately capturing the ambiguity present in these datasets. Our results show that it is indeed possible to exploit this ambiguity information, and that for the same amount of data, a model trained to recognize ambiguity shows signs of higher performance in the same task as well as in other downstream tasks.

However, our dataset was created using existing resources and lacks in quality and quantity. While it was enough to show that this research direction is promising, it limited the strength of our results. In future work, we wish to obtain larger amounts of data by using crowdsourcing techniques, and expand our scope to other NLP tasks as well.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 21H03502, JST PRESTO Grant Number JPMJPR20C4, and by the "la Caixa" Foundation (ID 100010434), under agreement LCF/BQ/AA19/11720042.

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain Natural Language Inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779, Online. Association for Computational Linguistics.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

John P Lalor, Hao Wu, and Hong Yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 648. NIH Public Access.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What Can We Learn from Collective Human Opinions on Natural Language Inference Data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7(0):677–694.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

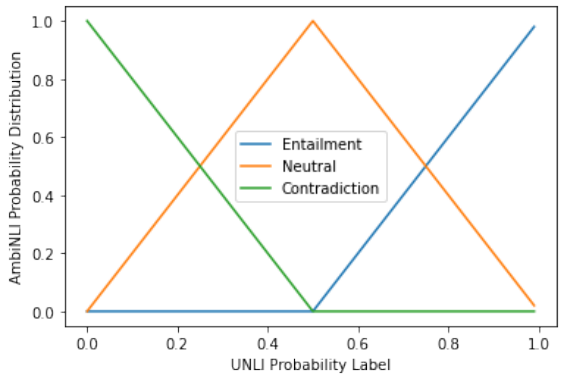


Figure 1: Linear approach to converting the UNLI regression value into an ambiguity distribution.

A Conversion Function

Figure 1 shows a plot of the linear conversion approach that we have taken to convert UNLI data into a probability distribution.

B Qualitative Analysis

To investigate the prediction differences between an ambiguous model and one trained on gold-labels, we compared AmbiSM and AmbiSM Gold predictions on the ChaosMNLI dataset (see Table 6). We use the new labels obtained from the ChaosNLI majority vote, instead of the original MNLI labels. We focus on two situations: 1) when only AmbiSM can predict the label correctly and 2) when only AmbiSM Gold can predict the label correctly. We picked samples from the high entropy regions to observe how the models deal with ambiguity. Generally, AmbiSM has a higher tendency towards neutrality. However, it was also able to show confidence in some samples that are more entailed or contradicted. On the other hand, we also observe some samples that were missed by AmbiSM due to its tendency, while AmbiSM Gold could predict them correctly.

Furthermore, we show the label counts for the samples that were correctly labeled by only one of the two models in Figure 2. The labels of the samples that are predicted correctly by AmbiSM Gold show the same distribution as the ChaosMNLI dataset as a whole. However, within the samples that are only predicted correctly by AmbiSM we can find a higher amount of neutral labels. This emphasizes that the behavior of the model trained on ambiguity targets can deal with neutral labels in NLI better; neutral labels are likely to be the biggest source of ambiguity.

Premise	Hypothesis	CHAOS	ASM	ASMG
Only AmbiSM is correct				
They were in rotation on the ground grabbing their weapons.	The woman rolled and drew two spears before the horse had rolled and broken the rest.	$E^{0.33}$ $\mathbf{N^{0.51}}$ $C^{0.16}$	$E^{0.178}$ $\mathbf{N^{0.522}}$ $C^{0.300}$	$E^{0.065}$ $N^{0.282}$ $\mathbf{C^{0.653}}$
Some of the unmet needs are among people who can pay, but who are deterred from seeking a lawyer because of the uncertainty about legal fees and their fear of the profession.	Some people can't afford it.	$\mathbf{E^{0.47}}$ $N^{0.40}$ $C^{0.13}$	$\mathbf{E^{0.572}}$ $N^{0.398}$ $C^{0.030}$	$E^{0.476}$ $\mathbf{N^{0.494}}$ $C^{0.030}$
This number represents the most reliable, albeit conservative, estimate of cases closed in 1999 by LSC grantees.	This is an actual verified number of closed cases.	$E^{0.21}$ $N^{0.12}$ $\mathbf{C^{0.67}}$	$E^{0.281}$ $N^{0.151}$ $\mathbf{C^{0.568}}$	$\mathbf{E^{0.485}}$ $N^{0.123}$ $C^{0.391}$
Only AmbiSM Gold is correct				
And it needs work too, you know, in case I have to jump out with this parachute from my lil' blue sports plane for real.'	It needs to work Incase he has to jump out a window.	$\mathbf{E^{0.44}}$ $N^{0.28}$ $C^{0.28}$	$E^{0.414}$ $\mathbf{N^{0.429}}$ $C^{0.156}$	$\mathbf{E^{0.489}}$ $N^{0.386}$ $C^{0.125}$
uh wasn't that Jane Eyre no he wrote Jane Eyre too	Was it Jane Eyre or not?	$\mathbf{E^{0.58}}$ $N^{0.36}$ $C^{0.06}$	$E^{0.398}$ $\mathbf{N^{0.422}}$ $C^{0.180}$	$\mathbf{E^{0.474}}$ $N^{0.413}$ $C^{0.113}$
Thus, the imbalance in the volume of mail exchanged magnifies the effect of the relatively higher rates in these countries.	There is an imbalance in ingoing vs outgoing mail.	$\mathbf{E^{0.60}}$ $N^{0.35}$ $C^{0.05}$	$E^{0.400}$ $\mathbf{N^{0.499}}$ $C^{0.101}$	$\mathbf{E^{0.458}}$ $N^{0.450}$ $C^{0.092}$

Table 6: Example of ChaosMNLI prediction for AmbiSM and AmbiSM Gold. **CHAOS** is the human distribution, **ASM** is the predicted distribution by AmbiSM and **ASMG** is the predicted distribution by AmbiSM Gold. The labels E, N, and C stand for entailment, neutral, and contradiction and their probabilities are appended.

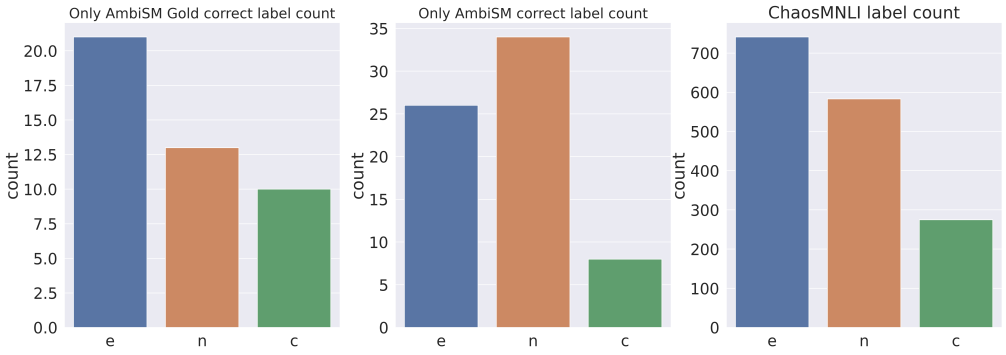


Figure 2: The count plot of the labels of the correctly predicted samples by either AmbiSM Gold or AmbiSM, AmbiSM Gold (left), AmbiSM (middle), and the labels of the whole ChaosMNLI (right). The labels e, n, and c stand for entailment, neutral, and contradiction respectively.

Modeling Discriminative Representations for Out-of-Domain Detection with Supervised Contrastive Learning

Zhiyuan Zeng[1*], **Keqing He**[2*], **Yuanmeng Yan**[1], **Zijun Liu**[1], **Yanan Wu**[1]
Hong Xu[1], **Huixing Jiang**[2], **Weiran Xu**[1*]
[1]Pattern Recognition & Intelligent System Laboratory
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
{zengzhiyuan,yanyuanmeng,liuzijun,yanan.wu,xuhong}@bupt.edu.cn
{hekeqing,jianghuixing}@meituan.com, {xuweiran}@bupt.edu.cn

Abstract

Detecting Out-of-Domain (OOD) or unknown intents from user queries is essential in a task-oriented dialog system. A key challenge of OOD detection is to learn discriminative semantic features. Traditional cross-entropy loss only focuses on whether a sample is correctly classified, and does not explicitly distinguish the margins between categories. In this paper, we propose a supervised contrastive learning objective to minimize intra-class variance by pulling together in-domain intents belonging to the same class and maximize inter-class variance by pushing apart samples from different classes. Besides, we employ an adversarial augmentation mechanism to obtain pseudo diverse views of a sample in the latent space. Experiments on two public datasets prove the effectiveness of our method capturing discriminative representations for OOD detection. [1]

1 Introduction

Detecting Out-of-Domain (OOD) or unknown intents from user queries is an essential component in a task-oriented dialog system (Gnewuch et al., 2017; Akasaki and Kaji, 2017; Tulshan and Dhage, 2018; Shum et al., 2018). It aims to know when a user query falls outside their range of predefined supported intents to avoid performing wrong operations. Different from normal intent detection tasks, we do not know the exact number of unknown intents in practical scenarios and can barely annotate extensive OOD samples. Lack of real OOD examples leads to poor prior knowledge about these unknown intents, making it challenging to identify OOD samples in the task-oriented dialog system.

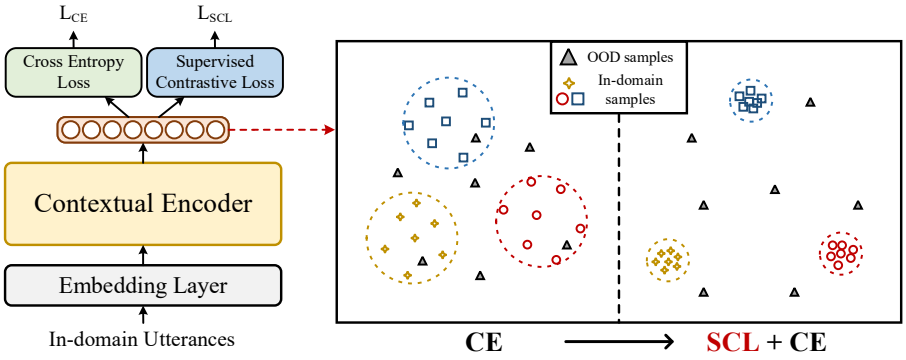
Previous methods of OOD detection can be generally classified into two types: supervised and unsupervised OOD detection. Supervised OOD detection (Scheirer et al., 2013; Fei and Liu, 2016; Kim and Kim, 2018; Larson et al., 2019; Zheng et al., 2020; Zeng et al., 2021b) represents that there are extensive labeled OOD samples in the training data. In contrast, unsupervised OOD detection (Bendale and Boult, 2016; Hendrycks and Gimpel, 2017; Shu et al., 2017; Lee et al., 2018; Ren et al., 2019; Lin and Xu, 2019; Xu et al., 2020; Zeng et al., 2021a) means no labeled OOD samples except for labeled in-domain data. Specifically, for supervised OOD detection, Fei and Liu (2016); Larson et al. (2019), form a $(N+1)$-class classification problem where the $(N+1)$-th class represents the unseen intents. Further, Zheng et al. (2020) uses labeled OOD data to generate an entropy regularization term to enforce the predicted distribution of OOD inputs closer to the uniform distribution. However, these methods heavily rely on large-scale time-consuming labeled OOD data. Compared to these supervised methods, unsupervised OOD detection first learns discriminative intent representations via in-domain (IND) data, then employs detecting algorithms, such as Maximum Softmax Probability (MSP) (Hendrycks and Gimpel, 2017), Local Outlier Factor (LOF) (Lin and Xu, 2019), Gaussian Discriminant Analysis (GDA) (Xu et al., 2020) to compute the similarity of features between OOD samples and IND samples. In this paper, we focus on the unsupervised OOD detection.

A key challenge of unsupervised OOD detection is to learn discriminative semantic features via IND data. We hope to cluster the same type of IND intents more tightly and separate different types of IND intents further. Traditional softmax loss (Hendrycks and Gimpel, 2017) only focuses on whether the sample is correctly classified, and does not explicitly distinguish the relationship between

[*]The first two authors contribute equally. Weiran Xu is the corresponding author.

[1]Our code is available at https://github.com/parZival27/supervised-contrastive-learning-for-out-of-domain-detection.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 870–878
August 1–6, 2021. ©2021 Association for Computational Linguistics



a) The in-domain intent classifier b) Using intent representations for OOD detection

Figure 1: The overall architecture of our proposed method. We first train an intent classifier on IND data using CE or SCL+CE objectives. Then, we extract the intent representation of a test sample to detect OOD.

categories. Further, Lin and Xu (2019) proposes a large margin cosine loss (LMCL) (Wang et al., 2018) which maximizes the decision margin in the latent space. LMCL forces the model to not only classify correctly but also maximize inter-class variance and minimize intra-class variance. Following the similar motivation, we aim to pull intents belonging to the same class together while simultaneously pushing apart samples from different classes to further model discriminative semantic features.

In this paper, we propose a supervised contrastive learning (SCL) model to learn discriminative semantic intent representation for OOD detection. SCL aims to minimize intra-class variance by pulling together IND intents belonging to the same class and maximize inter-class variance by pushing apart samples from different classes. Empirical results demonstrate the effectiveness of discriminative representation for OOD detection. Besides, to enhance the diversity of data augmentation in SCL, we employ an adversarial attack mechanism to obtain pseudo hard positive samples in the latent space by computing model-agnostic adversarial worst-case perturbations to the inputs. Our contributions are three-fold: (1) To the best of our knowledge, we are the first to apply supervised contrastive learning to OOD detection. (2) Compared to cross-entropy (CE) loss, SCL+CE can maximize inter-class variance and minimize intra-class variance to learn discriminative semantic representation. (3) Extensive experiments and analysis on two public datasets demonstrate the effectiveness of our method.

2 Methodology

Overall Architecture Fig 1 shows the overall architecture of our proposed method. As Fig 1(a) displays, we first train an IND intent classifier using CE or SCL+CE objectives in the training stage. Then in the test stage, we extract the intent feature of a test sample and employ the detection algorithms MSP (Hendrycks and Gimpel, 2017), LOF (Lin and Xu, 2019) or GDA (Xu et al., 2020) to detect OOD. [2] Fig 1(b) demonstrates the effectiveness of our method capturing discriminative intent representations, where SCL+CE can maximize inter-class variance and minimize intra-class variance.

Supervised Contrastive Learning We first review the classic cross-entropy (CE) loss and its improved version, large margin cosine loss (LMCL). Then we explain our supervised contrastive loss (SCL) in detail. Given an IND sample x_i and its intent label y_i, we adopt a BiLSTM (Hochreiter and Schmidhuber, 1997) or BERT (Devlin et al., 2019) encoder to get the intent representation s_i. The CE loss and LMCL are defined as follows [3]:

$$\mathcal{L}_{CE} = \frac{1}{N}\sum_{i} -\log \frac{e^{W_{y_i}^T s_i/\tau}}{\sum_j e^{W_j^T s_i/\tau}} \quad (1)$$

$$\mathcal{L}_{LMCL} = \frac{1}{N}\sum_{i} -\log \frac{e^{W_{y_i}^T s_i/\tau}}{e^{W_{y_i}^T s_i/\tau} + \sum_{j\neq y_i} e^{(W_j^T s_i + m)/\tau}} \quad (2)$$

where N denotes the number of training samples, y_i is the ground-truth class of the i-th sample, τ is the temperature factor, W_j is the weight vector of the j-th class, and m is the cosine margin. Compared to CE, LMCL adds a normalized decision margin on the negative classes and forces the model to explicitly distinguish positive class and negative classes. Our experiment 3.2 shows LMCL can slightly improve the performance of OOD detec-

[2] In this paper, we focus on the first training stage. Thus we dive into the details about the detection algorithms MSP, LOF and GDA in the appendix.

[3] For brevity, we omit the L2 normalization on both features and weight vectors for LMCL.

Models		CLINC-Full				CLINC-Small			
		IND		OOD		IND		OOD	
		ACC	F1	Recall	F1	ACC	F1	Recall	F1
LSTM	CE	86.34	87.73	63.72	65.23	84.24	84.30	60.40	61.07
	LMCL	86.83	87.90	64.14	65.79	84.46	84.87	60.72	61.89
	SCL+CE(ours)	87.01	88.28	66.80	67.68	85.73	86.61	63.96	64.44
	SCL+LMCL(ours)	**87.37**	**88.60**	**66.92**	**68.04**	**85.93**	**87.02**	**64.16**	**64.70**
BERT	CE	88.13	88.98	64.24	66.17	86.68	86.20	61.64	62.58
	LMCL	88.57	89.12	64.76	66.80	86.76	86.64	62.20	63.11
	SCL+CE(ours)	88.97	89.57	66.84	68.03	87.65	88.07	64.44	64.52
	SCL+LMCL(ours)	**89.20**	**90.03**	**67.28**	**68.21**	**87.87**	**88.30**	**64.64**	**65.01**

Table 1: Performance comparison on CLINC-Full and CLINC-Small datasets ($p < 0.05$ under t-test).

tion. To further model discriminative intent representations, motivated by recent contrastive learning work (Chen et al., 2020; He et al., 2020; Khosla et al., 2020; Gunel et al., 2020), we propose a supervised contrastive learning objective to minimize intra-class variance and maximize inter-class variance:

$$\mathcal{L}_{SCL} = \sum_{i=1}^{N} -\frac{1}{N_{y_i} - 1} \sum_{j=1}^{N} \mathbf{1}_{i \neq j} \mathbf{1}_{y_i = y_j} \log \frac{\exp\left(s_i \cdot s_j / \tau\right)}{\sum_{k=1}^{N} \mathbf{1}_{i \neq k} \exp\left(s_i \cdot s_k / \tau\right)} \quad (3)$$

where N_{y_i} is the total number of examples in the batch that have the same label as y_i and $\mathbf{1}$ is an indicator function. Note that we only perform SCL on the IND data since we focus on the unsupervised OOD detection where no labeled OOD data exists. As Fig 1(b) shows, SCL aims to pull together IND intents belonging to the same class and pushing apart samples from different classes, which helps recognize OOD intents near the decision boundary. In the implementation, we first pre-train the intent classifier using SCL, then finetune the model using CE or LMCL, both on the IND data. We compare iterative training and joint training in the appendix.

Adversarial Augmentation Chen et al. (2020) has proved the necessity of data augmentation for contrastive learning. However, there is no simple and effective augmentation strategy in the NLP area, which requires much handcrafted engineering. Thus, we apply adversarial attack (Goodfellow et al., 2015; Kurakin et al., 2016; Jia and Liang, 2017; Zhang et al., 2019; Yan et al., 2020) to generate pseudo positive samples to increase the diversity of views for contrastive learning. Specifically, we need to compute the worst-case perturbation $\boldsymbol{\delta}$ that maximizes the original cross-entropy loss $\mathcal{L}_{CE}$: $\boldsymbol{\delta} = \underset{\|\boldsymbol{\delta}'\| \leq \epsilon}{\arg\max} \; \mathcal{L}_{CE}\left(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta}'\right)$, where $\boldsymbol{\theta}$ represents the parameters of the intent classifier and $\boldsymbol{x}$ denotes a given sample. ϵ is the norm bound of the perturbation $\boldsymbol{\delta}$. We apply Fast Gradient Value (FGV) (Rozsa et al., 2016) to approximate the perturbation $\boldsymbol{\delta}$:

$$\boldsymbol{\delta} = \epsilon \frac{g}{||g||}; \text{where } g = \nabla_{\boldsymbol{x}} \mathcal{L}_{CE}(f(\boldsymbol{x}; \boldsymbol{\theta}), y) \quad (4)$$

We perform normalization to g and then use a small ϵ to ensure the approximate is reasonable. Finally, we can obtain the pseudo augmented sample $\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{\delta}$ in the latent space. The pseudo samples are applied to augment positive views per anchor in SCL. Ablation study 3.3 shows adversarial augmentation significantly improves the performance of SCL for OOD detection.

3 Experiments

3.1 Setup

Datasets We use two benchmark OOD datasets, CLINC-Full and CLINC-Small (Larson et al., 2019). We report IND metrics: Accuracy(Acc) and F1, and OOD metrics: Recall and F1. OOD Recall and F1 are the main evaluation metrics in this paper. **Baselines** We adopt LSTM and BERT as our intent classifier and compare SCL with CE and LMCL. Since only using SCL can't classify in-domain intents directly, we first pre-train the classifier using SCL, then finetune the model using CE or LMCL, both on the IND data. We use three OOD detection algorithms MSP, LOF and GDA to verify the generalization capability of SCL. We present dataset statistics, implementation details, and results on MSP and LOF in the appendix.

3.2 Main Results

Tab 1 displays the main results on GDA. Combining SCL and CE/LMCL significantly outperforms all the baselines, both on OOD and IND metrics. For OOD metrics, using SCL+CE in LSTM outperforms CE by 3.08%(Recall) and 2.45%(F1) on CLINC-Full, 3.56%(Recall) and 3.37%(F1) on CLINC-Small. Similar improvements based on

models		min	max	mean	median
LSTM	CE	1.13E-07	2.63E-04	4.23E-05	1.61E-05
	SCL+CE	4.35E-08	1.85E-04	3.23E-05	1.39E-05
BERT	CE	8.26E-08	2.23E-04	3.84E-05	1.56E-05
	SCL+CE	2.86E-08	1.67E-04	3.05E-05	1.36E-05

Table 2: Intra-class variance statistics.

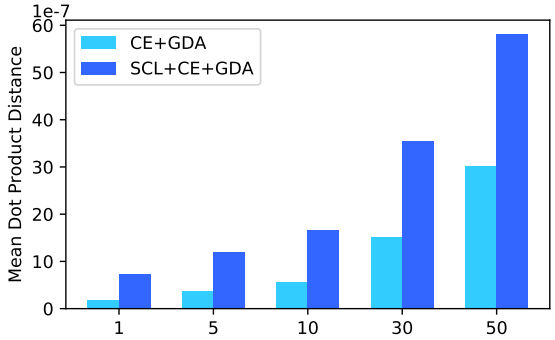


Figure 2: Comparison between inter-class distances.

LMCL are observed. The results prove the effectiveness of SCL for OOD detection. For IND metrics, using SCL+CE in LSTM outperforms CE by 0.67%(ACC) and 0.55%(F1) on CLINC-Full, 1.49%(ACC) and 2.31%(F1) on CLINC-Small. The results confirm SCL also helps IND intent detection. The difference between OOD and IND improvements is probably attributed to metric scale and data imbalance in the original test set. Besides, SCL gains higher improvements on CLINC-Small than CLINC-Full, which displays the advantage of our approach in the few-shot scenario (see details in Section 3.3). SCL also gets consistent improvements on BERT by 2.60%(Recall) and 1.86%(F1) on CLINC-Full OOD metrics, 0.84%(ACC) and 0.59%(F1) on CLINC-Full IND metrics, substantiating our method is model-agnostic for different OOD detection architectures.

3.3 Analysis

Analysis of IND feature distribution. We analyze the representation distribution of IND data on CLINC-Full dataset from two perspectives, intra-class and inter-class. We choose SCL+CE based on GDA to perform analysis. Tab 2 shows the statistics of intra-class variance, which can indicate the degree of clustering of intra-class data representations. Specifically, we average the variances of each sample normalized representation with the same intent label to its cluster center in the test set as cluster intra-class variance, then report min/max/mean/median values on all cluster intra-class variances. Results show SCL effectively decreases intra-class variances, especially in terms of max and mean values, which confirms SCL can converge intra-class intent representations.

Proportion		10%	20%	30%	40%	50%
IND F1	CE	63.31	70.77	77.84	81.55	84.30
	SCL+CE	69.50	75.14	81.45	84.18	86.61
	Relative↑	**9.78%**	**6.17%**	**4.64%**	**3.23%**	**2.74%**
OOD F1	CE	42.16	48.34	53.00	57.92	61.07
	SCL+CE	50.10	54.43	58.61	62.12	64.44
	Relative↑	**18.83%**	**12.60%**	**10.58%**	**7.25%**	**5.52%**

Table 3: Effect of training data size.

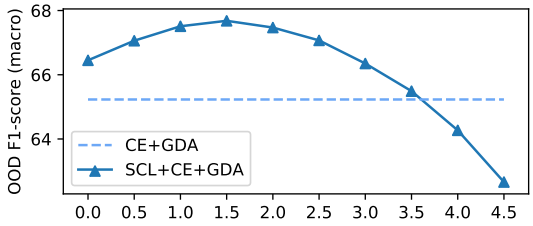


Figure 3: Effect of Adversarial Perturbation Norm.

Fig 2 shows the inter-class distances. We average dot product distances between each class center to its k nearest class centers, then average results of all classes as inter-class distance. The X-axis denotes the number of k. We observe a significant increase in SCL+CE compared to CE. When k is smaller, the increase is more obvious. It verifies SCL can maximize inter-class variance and distinguish intent classes. We also provide visualization analysis in the appendix. In summary, SCL can pull together IND intents belonging to the same class and push apart samples from different classes, which makes representations more discriminative.

Effect of IND Training Data Size. Tab 3 shows the effect of IND training data size. We randomly choose training data with a certain proportion from CLINC-Full IND data and use the original test set for evaluation. We use the LSTM+GDA setting. Results show SCL+CE consistently outperforms CE. Besides, with the decrease of training data size, the relative improvements gradually increase. It proves SCL has strong robustness for improving OOD detection, especially in the few-shot scenario.

Analysis of Adversarial Perturbation Norm. Fig 3 shows the effect of adversarial perturbation norm ϵ on OOD detection performance. We conduct the experiments on CLINC-Full dataset, using LSTM and GDA. The X-axis denotes the value of ϵ. The CE+GDA dashed line means no SCL pre-training and $\epsilon = 0.0$ in the SCL+CE+GDA solid line means no adversarial augmentation. In general, both SCL and adversarial augmentation contribute to the improvements and $\epsilon \in (1.0, 2.0)$ achieves better performances. Compared with the baseline without SCL, the SCL+CE method with a smaller adversarial perturbation can still obtain

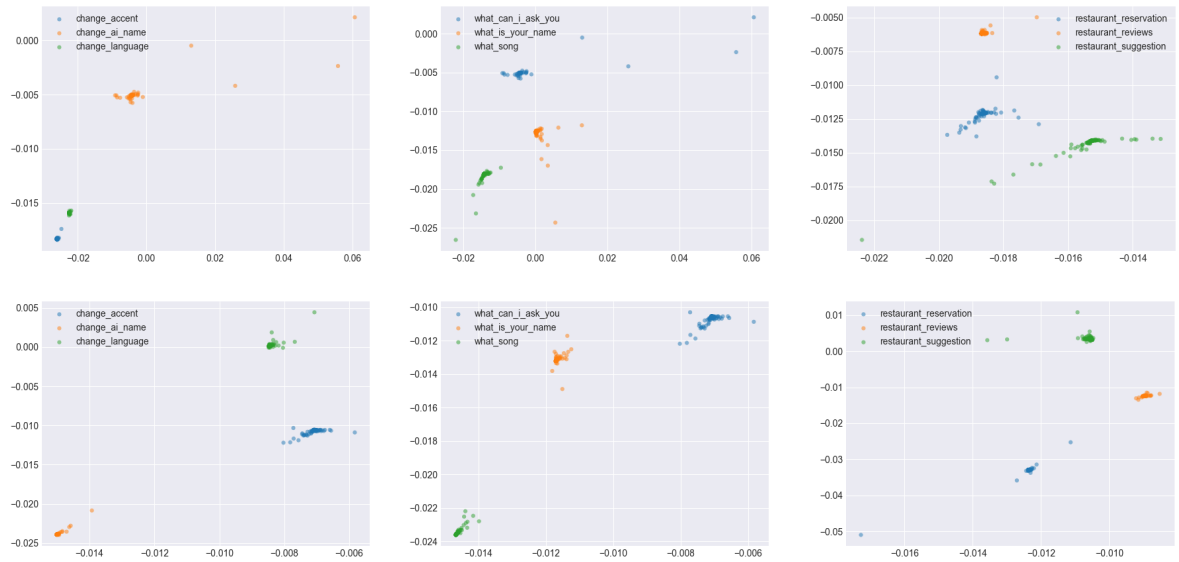


Figure 4: Visualization of in-domain representation distribution.

settings		IND		OOD	
dimension	batch size	ACC	F1	Recall	F1
128	50	87.01	88.28	66.80	67.68
128	100	87.52	88.60	67.08	68.12
128	200	88.10	89.05	67.56	68.63
256	50	87.17	88.40	66.96	67.92
256	100	87.85	88.96	67.32	68.37
256	200	88.37	89.24	67.76	68.94
512	50	87.35	88.78	67.32	68.47
512	100	88.14	89.22	67.64	68.69
512	200	88.54	89.50	68.00	69.27

Table 4: Parameter analysis of batch size and representation dimension.

better results but lower than the results with an optimal range of perturbation, while large norms tend to damage the effect of SCL. Our method still performs well with a broad range of adversarial perturbation and is insensitive to hyperparameters.

Parameter Analysis. As our proposed SCL is a method involving contrastive learning, we analyze batch sizes and representation dimensions to further verify the effectiveness, whose results are presented in Table 4. We conduct experiments in CLINC-Full dataset, using LSTM and SCL+CE objective for training and GDA for detection. With the increase of batch size and representation dimension, both in-domain and OOD metrics are slightly improved. However, compared with the method proposed in this paper, the improvement is relatively limited. In general, our proposed method is not sensitive to hyperparameters and can show the expected effect under a wide range of reasonable settings.

Feature Visualization. As shown in Fig 4, we extract several groups of similar classes for PCA visualization analysis. The three pictures in the upper part represent training using only CE, while the three pictures in the lower part use SCL+CE for training. In the same column, we sample the same classes for observation. It is worth noting that the scale of the image has been adjusted adaptively in order to display all the data. The actual distance can be sensed by observing the marking of the coordinate axis. After SCL is added, the distance between similar classes is significantly expanded, and the data in the same classes are more closely clustered.

4 Conclusion

In this paper, we focus on the unsupervised OOD detection where no labeled OOD data exist. To learn discriminative semantic intent representations via in-domain data, we propose a novel supervised contrastive learning loss (SCL). SCL aims to minimize intra-class variance by pulling together in-domain intents belonging to the same class and maximize inter-class variance by pushing apart samples from different classes. Experiments and analysis confirm the effectiveness of SCL for OOD detection. We hope to provide new guidance for future OOD detection work.

Acknowledgements

This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

Broader Impact

Task-oriented dialog systems have demonstrated remarkable performance across a wide range of applications, with the promise of a significant positive impact on human production mode and lifeway. However, in scenarios where information is complex and rapidly changing, models usually face input that is meaningfully different from typical examples encountered during training. Current models are prone to make unfounded predictions on these inputs, which may affect human judgment and thus impair the safety of models in practical applications. In domains with the greatest potential for societal impacts, such as navigation or medical diagnosis, models should be able to detect potentially agnostic OOD and be robust to high-entropy inputs to avoid catastrophic errors. This work proposes a novel unsupervised OOD detection method that using supervised contrastive learning to learn discriminative semantic intent representations. The effectiveness and robustness of the model are significantly improved by adding a supervised contrastive learning pre-training stage, which takes a step towards the ultimate goal of enabling the safe real-world deployment of task-oriented dialog systems in safety-critical domains. The experimental results have been reported on standard benchmark datasets for considerations of reproducible research.

References

Satoshi Akasaki and Nobuhiro Kaji. 2017. Chat detection in an intelligent assistant: Combining task-oriented and non-task-oriented spoken dialogue systems. *ArXiv*, abs/1705.00746.

Abhijit Bendale and Terrance E. Boult. 2016. Towards open set deep networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *HLT-NAACL*.

Ulrich Gnewuch, S. Morana, and A. Maedche. 2017. Towards designing cooperative and social conversational agents for customer service. In *ICIS*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *ArXiv*, abs/2011.01403.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ArXiv*, abs/1610.02136.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, A. Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *ArXiv*, abs/2004.11362.

Joo-Kyung Kim and Young-Bum Kim. 2018. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisficing false acceptance rates. *ArXiv*, abs/1807.00072.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *ArXiv*, abs/1807.03888.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. *ArXiv*, abs/1906.02845.

Andras Rozsa, Ethan M Rudd, and Terrance E Boult. 2016. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32.

Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. 2013. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1757–1772.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *EMNLP*.

H. Shum, X. He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19:10–26.

Amrita S. Tulshan and S. Dhage. 2018. Survey on virtual assistant: Google assistant, siri, cortana, alexa.

Feng Wang, Jian Cheng, Weiyang Liu, and H. Liu. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25:926–930.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1452–1460, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuanmeng Yan, Keqing He, H. Xu, Si hong Liu, Fanyu Meng, Min Hu, and Weiran Xu. 2020. Adversarial semantic decoupling for recognizing open-vocabulary slots. In *EMNLP*.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Hong Xu, and Weiran Xu. 2021a. Adversarial self-supervised learning for out-of-domain detection. In *NAACL*.

Zhiyuan Zeng, Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2021b. Adversarial generative distance-based classifier for robust out-of-domain detection. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7658–7662.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1198–1209.

A Dataset Details

Table 5 shows the details of two benchmark OOD dataset[4] CLINC-Full and CLINC-Small (Larson et al., 2019). They both contain 150 in-domain intents across 10 domains. It is worth noting that our paper does not use labeled OOD data from the training set in the training stage.

CLINC	Full	Small
Avg utterance length	9	9
Intents	150	150
Training set size	15100	7600
Training samples per class	100	50
Training OOD samples amount	100	100
Development set size	3100	3100
Development samples per class	20	20
Development OOD samples amount	100	100
Testing Set Size	5500	5500
Testing samples per class	30	30
Development OOD samples amount	1000	1000

Table 5: Statistics of the CLINC datasets.

B Baseline Details

We compare many types of unsupervised OOD detection models. Therefore, the model proposed in this paper can be divided into the training stage and detection stage. For each model LSTM or BERT, we use different detection methods to verify its performance. The innovation of this paper focuses mainly on the training stage. Due to the limitation of space, we do not detailed introduce the detection methods in the main body. We will supplement the relevant contents as follows:

MSP (Maximum Softmax Probability)(Hendrycks and Gimpel, 2017) applies a threshold on the maximum softmax probability where the threshold is set to 0.5 according to the dev set.

LOF (Local Outlier Factor)(Lin and Xu, 2019) uses the local outlier factor to detect unknown intents. The motivation is that if an example's local density is significantly lower than its k-nearest neighbor's, it is more likely to be considered as the unknown intents.

GDA (Gaussian Discriminant Analysis)(Xu et al., 2020) is a generative distance-based classifier for out-of-domain detection with Euclidean space. They estimate the class-conditional distribution on feature spaces of DNNs via Gaussian discriminant analysis (GDA) to avoid over-confidence problems and use Mahalanobis distance to measure the confidence score of whether a test sample belongs to OOD. GDA is the state-of-the-art detection methods till now, so we adopt GDA as our main detection algorithm. We also report MSP and LOF results in Section D.

C Implementation Details

We use the public pre-trained 300 dimensions GloVe embeddings (Pennington et al., 2014)[5] or bert-base-uncased (Devlin et al., 2019)[6] model to embed tokens. We use a single-layer BiLSTM as a feature extractor and set the dimension of hidden states to 128. The dropout value is fixed at 0.5. We use Adam optimizer (Kingma and Ba, 2014) to train our model. We set a learning rate to 1E-03 for GloVe+LSTM and 1E-04 for Bert. In the training stage, 100 epochs of supervised contrastive training are first conducted, then 10 epochs of finetune training are conducted with CE or LMCL. Both phases are training only on in-domain labeled data. The training stage has an early stop setting with patience equal to 5. We use the best F1 scores on the validation set to calculate the GDA threshold adaptively. Each result of the experiments is tested 5 times under the same setting and gets the average value. The norms of adversarial perturbation are obtained by the heuristic method, in which MSP and LOF are 1.0 and GDA is 1.5. The training stage of our model lasts about 10 minutes using GloVe embeddings, and 18 minutes using Bert-base-uncased, both on a single Tesla T4 GPU(16 GB of memory). The average value of the trainable model parameters is 3.05M.

D Supplementary Experimental Results

Various Detection Methods In this paper, the experiments and analysis are mainly conducted around the training stage. Different detection models are used to verify the generalization of our proposed method. Due to the limitation of space, we use GDA for most of the presentation in the main body. The main experiments of LOF and MSP using LSTM feature extractor are shown in Table 6. It is worth noting that using different detection methods can obtain the same analysis results as the main experimental in the main body.

Combining two training stages in different ways We display results of different combining

[4]https://github.com/clinc/oos-eval

[5]https://github.com/stanfordnlp/GloVe

[6]https://github.com/google-research/bert

Models		CLINC-Full				CLINC-Small			
		IND		OOD		IND		OOD	
		ACC	F1	Recall	F1	ACC	F1	Recall	F1
LOF	CE	85.46	85.80	57.40	58.78	82.45	82.73	52.88	53.90
	LMCL	85.87	86.08	58.32	59.28	82.83	82.98	53.96	54.63
	SCL+CE(ours)	86.52	86.80	60.72	61.80	83.13	83.39	56.88	57.48
	SCL+LMCL(ours)	**86.94**	**87.15**	**61.88**	**63.03**	**83.40**	**83.57**	**57.92**	**58.60**
MSP	CE	85.76	86.27	27.12	34.91	83.81	84.12	20.40	22.76
	LMCL	87.36	87.62	31.28	36.66	85.02	85.30	24.16	25.72
	SCL+CE(ours)	87.44	87.87	33.68	39.34	85.54	85.95	27.24	27.43
	SCL+LMCL(ours)	**88.89**	**89.21**	**35.40**	**41.75**	**86.87**	**87.20**	**29.28**	**31.02**

Table 6: Supplementary experimental results of LOF and MSP.

models	IND		OOD	
	ACC	F1	Recall	F1
CE	86.34	87.73	63.72	65.23
CE+SCL	82.29	83.59	61.96	63.40
multitask	86.69	88.02	65.76	67.25
SCL+CE	**87.01**	**88.28**	**66.80**	**67.68**

Table 7: Results of combining two training stages in different ways

ways of two training stages on CLINC-Full dataset using LSTM and GDA detection method in Table 7. CE is the baseline that only uses the cross-entropy loss function to train the feature extractor. SCL+CE follows the paradigm of pre-training first and then finetuning, which achieves the best performance. Besides, we try two different combinations to explore the relationship between the two training stages. CE+SCL means that we first conduct training to minimize cross-entropy loss, and then conduct supervised contrastive learning. The results show that the subsequent SCL leads to a decline in metrics, especially on in-domain. This is because SCL, while optimizing the representation distribution, compromises the mapping relationship with labels. Multitask means to optimize two losses simultaneously. This setting leads to mutual interference between two tasks, which affects the convergence effect and damages the performance and stability of the model. In general, SCL should be used as a pre-training method and CE as a finetuning method. The best results can be achieved by first using SCL to learn discriminative representation and then finetuning the model by CE.

Preview, Attend and Review: Schema-Aware Curriculum Learning for Multi-Domain Dialog State Tracking

Yinpei Dai[†], **Hangyu Li**[†], **Yongbin Li**[†]*, **Jian Sun**[†], **Fei Huang**[†], **Luo Si**[†], **Xiaodan Zhu**[‡]
[†]Alibaba Group
[‡]Ingenuity Labs Research Institute & ECE, Queen's University
{yinpei.dyp,hangyu.lhy,shuide.lyb}@alibaba-inc.com
{jian.sun,f.huang,luo.si}@alibaba-inc.com, zhu2048@gmail.com

Abstract

Existing dialog state tracking (DST) models are trained with dialog data in a random order, neglecting rich structural information in a dataset. In this paper, we propose to use curriculum learning (CL) to better leverage both the curriculum structure and schema structure for task-oriented dialogs. Specifically, we propose a model-agnostic framework called **S**chema-**a**ware **C**urriculum **L**earning for Dial**og** State Tracking (SaCLog), which consists of a preview module that pre-trains a DST model with schema information, a curriculum module that optimizes the model with CL, and a review module that augments mispredicted data to reinforce the CL training. We show that our proposed approach improves DST performance over both a transformer-based and RNN-based DST model (TripPy and TRADE) and achieves new state-of-the-art results on WOZ2.0 and MultiWOZ2.1.

1 Introduction

Dialog state tracking (DST) extracts users' goals in task-oriented dialog systems, where dialog states are often represented in terms of a set of slot-value pairs (Williams et al., 2016; Eric et al., 2020). Due to the language variety of multi-turn dialogs, the concepts of slots and values are often indirectly expressed in the conversation (such as co-references, ellipsis, and diverse appearances), which are a major bottleneck for improving DST performance (Gao et al., 2019; Hu et al., 2020). Many existing DST methods have focused on designing better model architectures to tackle the problems (Dai et al., 2018; Wu et al., 2019; Kim et al., 2020), but still neglect the full exploitation of two important aspects of structural information.

The first is *curriculum structure* in a dataset. Such a structure relies on a measure of the difficulty of examples, which can be used to guide the model training in an easy-to-hard manner, imitating the meaningful learning order in human curricula. This paradigm is called curriculum learning (CL) (Bengio et al., 2009) and has been shown useful in various other problems (Wang et al., 2021). DST training examples also vary greatly in their difficulty levels. As shown in Figure 1, for the same slot *'taxi-departure'*, a user can either inform its value *'nandos'* explicitly in a simple utterance or convey her intention implicitly via multi-round interactions, requiring a complex inference process to find the value *'golden house'* referred by the slot *'restaurant-name'*. However, CL has been rarely studied in DST, and models are often trained with dialog data in a random order.

Figure 1: An easy and a hard dialog example for DST.

In addition, *schema structure* is prominent in multi-domain task-oriented dialogs. A schema is specified by a collection of all possible slots and their values, which describes semantic relations among them. Some previous work utilized the structure via an extra schema graph in a regular training process (Chen et al., 2020; Zhu et al., 2020; Wu et al., 2020). We propose to incorporate schema information into CL through a pre-curriculum process, in which a DST model can be pre-trained with schema-related objectives to prepare for upcom-

*Corresponding author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 879–885
August 1–6, 2021. ©2021 Association for Computational Linguistics

ing DST examples. To reinforce the CL training, we can also expand those examples with frequent mispredictions during CL based upon the schema, enabling the model to accumulate more experience and perform better on similar cases.

Built on these motivations, we propose a novel framework named as **S**chema-**a**ware **C**urriculum **L**earning for Dial**og** State Tracking (SaCLog), which consists of three components: 1) a **preview module** that pre-trains the base part of a DST model (e.g., BERT and RNN) with objectives capturing the connections between the schema and dialog contexts, 2) a **curriculum module** that organizes training data from easy to hard and optimizes the model with CL, and 3) a **review module** which leverages schema-based data augmentation to extend mispredicted data to boost the CL training process further. The proposed approach is model-agnostic, in the sense that it can be incorporated into different DST models. To the best of our knowledge, this is the first attempt to apply CL to the DST task. We show that our proposed approach improves DST performance over both a transformer-based and RNN-based DST model (TripPy and TRADE) and achieves new state-of-the-art results on WOZ2.0 and MultiWOZ2.1.

2 Problem Formulation

We denote a dialog context containing t turns as $X_t = \{(R_1, U_1), ..., (R_t, U_t)\}$, where R_i and U_i represent system and user utterance at the i-th turn respectively. DST is tasked to extract turn-level or discourse-level dialog states in the form of a set of slot-value pairs given X_t. A turn-level dialog state $Y_t = \{(s, v_t), s \in \mathcal{S}\}$ is the slot-value pairs extracted only from (R_t, U_t) at current turn t, where $\mathcal{S}$ is a predefined set of slot s in the schema and v_t is the corresponding value[1] of the slot s. A discourse-level dialog state Z_t is the accumulation of L_t, representing all slot-value pairs that have been expressed over the course of the dialog until the t-th turn. We denote a dialog data for DST as $d_t = \{X_t, Y_t, Z_t\}$ and the training dataset as $\mathcal{D}$.

3 Schema-Aware Curriculum Learning

In this section, we first introduce the core curriculum module about how to apply the basic CL to the DST task; we then describe the preview and review module, which exploit the schema structure to facilitate the CL training process. The overall framework of SaCLog is shown in Figure 2.

3.1 Curriculum Learning for DST

We propose curriculum learning for DST and design two sub-modules: a **difficulty scorer** that measures the difficulty level of a dialog example with respect to a DST model, as well as a **training scheduler** module that arranges the scored data as a sequence of easy-to-hard training stages.

3.1.1 The Difficulty Scorer

As a dialog example could be intuitively *complex* for humans or inherently *difficult* for neural networks (NNs), both model-based and rule-based scores should be considered. We propose to use a hybrid scoring function that combines the advantages of model predictions and rules.

For model-based difficulty, we predict scores in a cross-validation-like manner. We divide $\mathcal{D}$ into K equal-sized subsets, where $K-1$ subsets are used to train a DST model to predict the remaining one. This process is repeated K times until every subset is predicted. The score $r_t^{mod} \in [0, 1]$ is computed based on the average accuracy of all mentioned slots (whose values are not *none*) in Y_t for each d_t. In our experiment, we train six models with the same architecture and different initialization seeds to obtain the mean value $\overline{r}_t^{mod}$ of model scores.

For rule-based difficulty, we consider 4 factors to fuse human prior knowledge about DST into our curriculum design: 1) current dialog turn number t; 2) the total token number of (R_t, U_t); 3) the number of mentioned name entities like 'hotel names' in Z_t; 4) the number of newly added or changed slots in Y_t. We set the maximum values of above factors as 7/50/4/6 respectively, and normalize all factors into $r_t^{rul,i} \in [0, 1]$, where i indicates the i-th factor.

Finally, the hybrid difficulty score is calculated jointly as $r_t^{hyb} = \alpha_0 \overline{r}_t^{mod} + \sum_{i=1}^{4} \alpha_i r_t^{rul,i}$, where $r^{hyb} \in [0, 1]$ and $\sum_{i=0}^{4} \alpha_i = 1$.

3.1.2 The Training Scheduler

We adopt a widely used strategy called *baby step* (Spitkovsky et al., 2010) to organize the scored data for CL. Specifically, we divide the score uniformly into N intervals and distribute the sorted data into N buckets accordingly. The optimization starts from the easiest bucket as the initial training stage. After reaching a fixed number of maximum epochs or convergence, the next bucket is merged

[1]Each s contains two special values, *none* and *dontcare*, indicating s has no values and can take any values respectively.

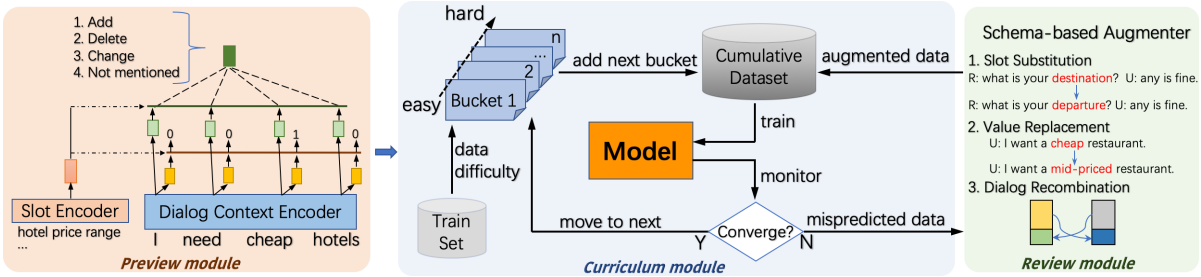


Figure 2: An overview of the SaCLog training procedures.

into the current training subset and shuffled for the next training stage. In our experiment, we set the maximum number of epochs as 3, and treat as the convergence if the training loss ceases to decrease and the loss value is within a threshold 15 for 100 steps. As the subset accumulates until all buckets are aggregated, we then continue to train the model for several extra epochs.

3.2 The Preview Module

In human learning, previewing learning materials helps develop an overall picture of what will be covering and can bring benefits to the learning process. In our task here, we propose new pre-training objectives to learn structural inductive bias of the schema structure. Specifically, our *preview* module contains a slot encoder to compute a slot embedding e_s for each input slot s, and a dialog context encoder to extract the hidden states of X_t as $E_t = [e_t^1, e_t^2, ...]$, then we have:

$$\begin{aligned} B_t^s &= [\phi_1^{sig}(e_s \oplus e_t^1), \phi_1^{sig}(e_s \oplus e_t^2), ...] \\ c_t^s &= \phi_4^{sft}(e_s \oplus \text{Att}(e_s, E_t)) \end{aligned} \quad (1)$$

where $\text{Att}(k, V)$ is the attention function using the vector k to query the vector sequence V to get a context vector and $\oplus$ the vector concatenation. $\phi_d^{sig}(\cdot)$ and $\phi_d^{sft}(\cdot)$ denote an FNN with one hidden layer having the same size as input layer, where the output layer is of size d, and is sigmoid and softmax respectively. B_t^s is a binary sequence indicates which span of X_t belongs to the value of s, while c_t^s is the classification logits indicates whether s is added, deleted, changed, or not mentioned in Y_t.

Therefore, for each slot s, we have a binary sequence loss L_{seq} and a classification loss L_{cls} to optimize. Such pre-training objectives help the encoders understand how a slot is roughly operated in the current dialog context and connected with all possible tokens regarding its values in the schema. The dialog context encoder is used for the parameter initialization of the base part of a DST model. The pre-trained corpus is constructed from MultiWOZ2.1 dialogs (Eric et al., 2020) and the off-the-shelf synthesized dialogs (Campagna et al., 2020), which contains 337,346 dialog data in total.

We also leverage the language modelling (LM) loss as an auxiliary loss L_{aux} to learn contextual representations of natural language. To be specific, we use the MLM loss (Devlin et al., 2019) as L_{aux} for transformer-based DST modes and the summation of both forward and backward LM losses (Peters et al., 2018) for RNN-based DST models. We only use the original MultiWOZ2.1 dialogs to optimize L_{aux}, considering that synthesized data is not suitable for natural language modelling. However, both the original and synthesized data are used to optimize L_{seq} and L_{cls}.

3.3 The Review Module

The process of review often help a learner consolidate difficult concepts newly learned. We design a *review* module to consider mispredicted examples as the concepts that the DST model has not grasped during CL, and utilize a schema-based data augmenter to produce similar cases from the examples. Specifically, the DST model is monitored at each stage of the CL training process. If a model is not converged at the end of an epoch in a training stage, we choose the top 10% incorrectly predicted examples according to their training losses as the resource to enlarge the cumulative dataset. The schema-based data augmenter uses three practical techniques to generate data as follows:

Slot Substitution. A mentioned slot name in (R_t, U_t) is changed into another slot name when its value is *dontcare*. Specifically, we first collect a word set for each slot name, e.g. {*'arrive', 'arriving', 'arrived'*} for the slot *'taxi-arriveby'*. Then, for a dialog data d_t where Y_t contains a slot s with the value *dontcare*, we substitute the word of s in the utterance with some word of another slot s' that is of the same domain and not mentioned in Y_t.

Value Replacement. A slot's value is replaced with another proper one when the value is explicitly contained in U_t. Specifically, we leverage the predefined schema in the dialog dataset to produce a value set for each slot and use the label map in (Heck et al., 2020) to figure out the position of value span within the utterance. The target value is then replaced with another one of the same slot.

Dialog Recombination. To recombine the dialog data d_t, we randomly search another dialog data in D that possesses the same mentioned slots (whose values are not *none*) in Y_t. We then cut and stitch their history X_{t-1} and current utterances (R_t, U_t), and exchange their Y_t to produce two new dialog data.

4 Experiments

Two popular datasets, WOZ2.0 (Wen et al., 2017) and MultiWOZ2.1 (Eric et al., 2020), are used to verify our approach. WOZ2.0 is a single-domain dataset with 1,200 dialogs and 3 slots. MultiWOZ2.1 is a multi-domain dialog dataset with 10,438 dialogs, where there are 30 slots spanning 7 domains. The data splits (train/valid/test) of WOZ2.0 and MultiWOZ2.1 are 600/200/400 and 8438/1000/1000, respectively. We use the joint goal accuracy (JGA), the ratio of dialog data whose Z_t is correct, as the evaluation metric. We apply SaCLog onto TripPy (Heck et al., 2020), a transformer-based DST model, and TRADE (Wu et al., 2019), an RNN-based DST model, to show its effect. The slot encoder and the dialog context encoder are weight-shared. We use a BERT_{base} as the encoder and the `[CLS]` embedding as the slot embedding in TripPy, and use a bi-GRU as the encoder and the concatenation of the first and last hidden state as the slot embedding for TRADE. We also follow TripPy to add 2 new slot operations (i.e. *refer/dontcare*) into the classification types of L_{cls}.

Implementation Details. For the preview module, we use Adam (Kingma and Ba, 2015) with a fixed learning rate 3e-5 for 3 epochs in the pre-training. The batch size for L_{aux} is 14 and the batch size for L_{seq} and L_{cls} is 64. For the curriculum module, we perform a warm-up strategy for Adam optimizer with a maximum learning rate 1e-4. Before CL, we train models on full dataset for 2 epochs. After all subsets are accumulated, we then train for 10 extra epochs with a minimum learning rate 1e-6. We set the bucket number $N = 10$ and the crossed fold $K = 5$. The batch size is 36 and the maximum length is 256. To simplify the review process, we conduct data augmentation after the CL training is finished.

Models	MultiWOZ2.1	WOZ2.0
GLAD (Zhong et al., 2018)	35.57%**	88.1±0.4%
SUMBT (Lee et al., 2019)	46.65%**	91.0±1.0%
DST-picklist (Zhang et al., 2019)	53.30%	–
Trippy (Heck et al., 2020)	55.29±0.28%	92.7±0.2%
SimpleTOD(Hosseini-Asl et al., 2020)	55.72%	–
CHAN (Shan et al., 2020)	58.55%	–
TripPy + ConvBERT	58.70%	93.1±0.3%*
TripPy + CoCoAug	60.53%	–
TripPy + SaCLog	**60.61±0.31%**	**94.2±0.2%**

Table 1: DST Results on MultiWOZ2.1 and WOZ2.0 in JGA. * Our implemetation. ** MultiWOZ2.0 results.

Models	JGA
TripPy (ours)	58.17±0.25%
+ CL (rule-based)	58.38±0.17%
+ CL (model-based)	58.71±0.21%
+ CL (hybrid)	**58.85±0.23%**
+ SaCLog (w/o. review)	60.19±0.26%
+ SaCLog (w/o. preview)	60.23±0.34%
+ SaCLog	**60.61±0.31%**

Table 2: Ablation results on MultiWOZ2.1. +CL means adding the curriculum module only.

4.1 Performance of TripPy+SaCLog

Tables 1 shows the results of our approach comparing to various baselines. Based upon TripPy, we obtain state-of-the-art performance on both datasets with SaCLog. The two closest baselines[2], ConvBERT (Mehri et al., 2020) and CoCoAug (Li et al., 2021), are also built upon TripPy, where ConvBERT enhances its performance by using external large-scale conversational corpora to pre-train a BERT_{base} and CoCoAug leverages a delicate counter-factual augmentation skill to produce much larger training data. Our method, however, benefits from the CL framework and improves TripPy by utilizing the preview and review modules.

Ablation Study To examine how SaCLog facilitates DST training, we conduct detailed ablation experiments on MultiWOZ2.1, as shown in Table 2. In our re-implementation, we improve the basic TripPy by around 3% JGA via training for longer epochs (30 vs.10) and pre-training a BERT_{base} on MultiWOZ2.1 corpus only with the MLM loss. First, we investigate the influence of

[2]We implemented SaCLog upon these two methods, but no significant gains are observed. We conjecture that this is due to SaCLog has already largely exploited TripPy's potential so that the additional improvement of the two methods is limited.

Model	MultiWOZ2.1	WOZ2.0
TRADE	45.6%*	88.3±0.6%
+ SaCLog	**49.3±0.5%**	**91.1±0.4%**

Table 3: Results of TRADE+SaCLog on MultiWOZ2.1 and WOZ2.0. * Reported in (Eric et al., 2020).

difficulty scores by adding the curriculum module and utilizing the same pre-trained BERT_{base}. As we can see, using the hybrid difficulty score achieves better JGA (58.85%) than using either single score, indicating that both model prediction and human knowledge are necessary. When incorporating the other two modules in the CL framework, the performance is greatly boosted further. The combination of both modules increases the JGA by 1.76%, suggesting that the schema-aware pre-training and dialog augmentation are crucial for improving DST performance in the CL training.

4.2 Performance of TRADE+SaCLog

We also apply SaCLog to the classical RNN-based generative DST model, TRADE. As Table 3 shows, SaCLog improves TRADE by around 3∼4% JGA on both datasets, demonstrating the effectiveness of SaCLog on different types of base DST models.

5 Related Work

Curriculum Learning (CL) has attracted increasing research interests in various NLP tasks, such as machine translation (Liu et al., 2020; Zhou et al., 2020), general language understanding (Xu et al., 2020), reading comprehension (Tay et al., 2019) and open-domain chatbots (Bao et al., 2020; Cai et al., 2020; Su et al., 2020). Yet, the research on using CL in task-oriented dialog systems is limited. There has been some work (Saito, 2018; Zhao et al., 2021) on using CL in dialog policy learning, but applying CL to DST has not been investigated.

Learning a structural inductive bias during pre-training has been shown beneficial in downstream tasks that require parsing semantics, such as text-to-SQL (Yu et al., 2021) and table cell recognition (Wang et al., 2020). There are also many works (Hou et al., 2018; Yoo et al., 2020; Yin et al., 2020) on dialog augmentation. We aim to integrate these methods to build a general CL framework for DST.

6 Conclusion

In this paper, we propose a model-agnostic framework named as schema-aware curriculum learning for DST, which exploits both the curriculum structure and the schema structure in task-oriented dialogs and shows to substantially improve DST performances. In the future, we plan to investigate CL approaches on other dialog modeling tasks.

Acknowledgments

The research of the last author is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2020. Plato-2: Towards building an open-domain chatbot via curriculum learning. *arXiv preprint arXiv:2006.16779*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Hengyi Cai, Hongshen Chen, Cheng Zhang, Yonghao Song, Xiaofang Zhao, Yangxi Li, Dongsheng Duan, and Dawei Yin. 2020. Learning from easy to complex: Adaptive multi-curricula learning for neural dialogue generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7472–7479.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Yinpei Dai, Zhijian Ou, Dawei Ren, and Pengfei Yu. 2018. Tracking of enriched dialog states for flexible conversational information access. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6139–6143. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Thirtyforth Conference on Neural Information Processing Systems*.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2020. SAS: Dialogue state tracking via slot attention and slot information sharing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6366–6375, Online. Association for Computational Linguistics.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. *International Conference on Learning Representations*.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Atsushi Saito. 2018. Curriculum learning based on reward sparseness for deep reinforcement learning of task completion dialogue management. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 46–51, Brussels, Belgium. Association for Computational Linguistics.

Yong Shan, Zekang Li, Jinchao Zhang, Fandong Meng, Yang Feng, Cheng Niu, and Jie Zhou. 2020. A contextual hierarchical attention network with adaptive objective for dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6322–6333, Online. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759, Los Angeles, California. Association for Computational Linguistics.

Yixuan Su, Deng Cai, Qingyu Zhou, Zibo Lin, Simon Baker, Yunbo Cao, Shuming Shi, Nigel Collier, and Yan Wang. 2020. Dialogue response selection with hierarchical curriculum learning. *59th Annual Meeting of the Association for Computational Linguistics*.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for read-

ing comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2020. Structure-aware pre-training for table understanding with tree-based transformers. *arXiv preprint arXiv:2010.12537*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Peng Wu, Bowei Zou, Ridong Jiang, and AiTi Aw. 2020. GCDST: A graph-based and copy-augmented multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1063–1073, Online. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialog state tracking with reinforced data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9474–9481.

Kang Min Yoo, Hanbit Lee, Franck Dernoncourt, Trung Bui, Walter Chang, and Sang-goo Lee. 2020. Variational hierarchical dialog autoencoder for dialog state tracking data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3406–3425, Online. Association for Computational Linguistics.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. *International Conference on Learning Representations*.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Yangyang Zhao, Zhenyu Wang, and Zhenhua Huang. 2021. Automatic curriculum learning with over-repetition penalty for dialogue policy learning. *AAAI*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

Su Zhu, Jieyu Li, Lu Chen, and Kai Yu. 2020. Efficient context and schema fusion networks for multi-domain dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 766–781, Online. Association for Computational Linguistics.

On the Generation of Medical Dialogs for COVID-19

Meng Zhou[*], Zechen Li[*], Bowen Tan[†], Guangtao Zeng[*], Wenmian Yang[*], Xuehai He[*], Zeqian Ju[*], Subrato Chakravorty[*], Shu Chen[*], Xingyi Yang[*], Yichen Zhang[*], Qingyang Wu[*], Zhou Yu[◇], Kun Xu[•], Eric Xing[†‡] and Pengtao Xie[*]
UC San Diego[*], CMU[†], Columbia University[◇], Tencent AI Lab[•],
Mohamed bin Zayed University of Artificial Intelligence[‡]
p1xie@eng.ucsd.edu

Abstract

Under the pandemic of COVID-19, people experiencing COVID19-related symptoms have a pressing need to consult doctors. Because of the shortage of medical professionals, many people cannot receive online consultations timely. To address this problem, we aim to develop a medical dialog system that can provide COVID19-related consultations. We collected two dialog datasets – CovidDialog – (in English and Chinese respectively) containing conversations between doctors and patients about COVID-19. While the largest of their kind, these two datasets are still relatively small compared with general-domain dialog datasets. Training complex dialog generation models on small datasets bears high risk of overfitting. To alleviate overfitting, we develop a multi-task learning approach, which regularizes the data-deficient dialog generation task with a masked token prediction task. Experiments on the CovidDialog datasets demonstrate the effectiveness of our approach. We perform both human evaluation and automatic evaluation of dialogs generated by our method. Results show that the generated responses are promising in being doctor-like, relevant to conversation history, clinically informative and correct. The code and the data are available at https://github.com/UCSD-AI4H/COVID-Dialogue.

1 Introduction

During the COVID-19 pandemic, people who are experiencing symptoms similar to those of COVID-19 or were exposed to risk factors have a pressing need to consult doctors. However, medical professionals are highly occupied, who do not have enough bandwidth to provide COVID19-related consultations.

To address this issue, we aim to develop a COVID19-targeted dialog system. We build two medical dialog datasets that contain conversations between doctors and patients, about COVID-19 and other pneumonia: (1) an English dataset containing 603 consultations, 1232 utterances, and 90664 tokens (English words); (2) a Chinese dataset containing 1088 consultations, 9494 utterances, and 406550 tokens (Chinese characters).

While the largest of their kind, these two datasets are still relatively small compared with general-domain dialog datasets. Training complex dialog generation models on small datasets bears high risk of overfitting. To alleviate overfitting in COVID-19 dialog generation, we develop a multi-task learning approach where a masked-token prediction (MTP) (Devlin et al., 2018) task is used to regularize the training of dialog generation models. Our method performs the MTP task and the dialog generation task simultaneously. The MTP loss serves as a regularization term and is optimized jointly with the dialog generation loss. Due to the presence of the MTP task, the dialog generation model is less likely to be biased to the dialog generation task defined on the small-sized training data. We perform experiments on our collected two COVID-19 dialog datasets, where the results demonstrate the effectiveness of our approach. We perform human evaluation and automatic evaluation of dialogs generated by our approach. The results show that the generated responses demonstrate high potential to be doctor-like, relevant to patient history, clinically informative and correct.

The major contributions of this paper are:

- We collect two medical dialog datasets about COVID-19: one in English, the other in Chinese.
- We develop a multi-task learning approach, which uses a masked-token prediction task to regularize the dialog generation task to alleviate overfitting.
- We evaluate our method on the collected COVID-19 dialog datasets and the results demonstrate the

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 886–896
August 1–6, 2021. ©2021 Association for Computational Linguistics

effectiveness of our method.

2 Related Works

Several works have studied data-driven medical dialog generation. Wei et al. (2018) proposed a task-oriented dialog system to make medical diagnosis automatically based on reinforcement learning. The system converses with patients to collect additional symptoms beyond their self-reports. Xu et al. (2019) proposed a knowledge-routed relational dialog system that incorporates medical knowledge graph into topic transition in dialog management. Xia et al. proposed an automatic diagnosis dialog system based on reinforcement learning. In these works, the neural models are trained from scratch on small-sized medical dialog datasets, which are prone to overfitting.

3 Datasets

We collected two dialog datasets – CovidDialog-English and CovidDialog-Chinese – which contain medical conversations between patients and doctors about COVID-19 and other related pneumonia. The statistics of these two datasets are summarized in Table 1.

The English Dataset The CovidDialog-English dataset contains 603 English consultations about COVID-19 and other related pneumonia, having 1,232 utterances. The number of tokens (English words) is 90,664. The average, maximum, and minimum number of utterances in a conversation is 2.0, 17, and 2 respectively. The average, maximum, and minimum number of tokens in an utterance is 49.8, 339, and 2 respectively. The conversations are from 582 patients and 117 doctors. Each consultation starts with a short description of the medical conditions of a patient, followed by the conversation between the patient and a doctor.

The Chinese Dataset The CovidDialog-Chinese dataset contains 1,088 Chinese consultations about COVID-19 and other related pneumonia, having 9,494 utterances. In this work, we develop models directly on Chinese characters without performing word segmentation. Each Chinese character in the text is treated as a token. The total number of tokens in the dataset is 406,550. The average, maximum, and minimum number of utterances in a conversation is 8.7, 116, and 2 respectively. The average, maximum, and minimum number of tokens in an utterance is 42.8, 2001, and 1 respectively. The conversations are from 935 patients and 352 doctors. Each consultation consists of three parts: (1) description of patient's medical condition and history; (2) conversation between patient and doctor; (3) (optional) diagnosis and treatment suggestions given by the doctor. In the description of the patient's medical condition and history, the following fields are included: present disease, detailed description of present disease, what help is needed from the doctor, how long the disease has been, medications, allergies, and past diseases. This description is used as the first utterance from the patient.

	English	**Chinese**
#dialogs	603	1,088
#tokens	90,664	406,550
Average #utterances per dialog	2.0	8.7
Max #utterances per dialog	17	116
Min #utterances per dialog	2	2
Average #tokens per utterance	49.8	42.8
Max #tokens per utterance	339	2,001
Min #tokens per utterance	2	1

Table 1: Statistics of the English and Chinese dialog datasets about COVID-19.

For both datasets, the dialogs are crawled from openly accessible medical websites whose owners make these dialogs visible to the public. The patients' personal information is de-identified by owners of these websites. We further checked the crawled dialogs to ensure they do not contain private information of patients. Besides, we also manually removed borderline sensitive information, such as specific dates and destinations in patients' travel histories. Experts in privacy and security domains helped to check the final version of shared data and ensured there is no breach of patient privacy or confidentiality.

4 Method

Given a dialog containing a sequence of alternating utterances between patient and doctor, we process it into a set of pairs $\{(s_i, t_i)\}$ where the target t_i is a response from the doctor and the source s_i is the conversation history – the concatenation of all utterances (from both patient and doctor) before t_i. A dialog generation model takes s as input and generates t. The model consists of an encoder which encodes s and a decoder which takes the encoding of s as input and generates t. The size of the CovidDialog datasets is small. Training neural dialog models on these small datasets has high risk of overfitting.

To solve this problem, we develop a multi-task

Split	# dialogs	# utterances	# pairs
Train	482	981	490
Validation	60	126	63
Test	61	122	61

Table 2: English dataset split statistics

	C	R	I	D
Transformer	2.24	2.57	2.53	2.29
GPT-2	2.58	2.91	2.65	3.09
BART	2.61	3.01	2.74	3.42
BART+TAPT	2.65	3.04	2.68	3.38
Ours	**2.83**	**3.16**	**2.88**	**3.47**
Groundtruth	3.26	3.63	3.51	3.55

Table 3: Human evaluation on the CovidDialog-English test set. C, R, I, and D represent correctness, relevance, informativeness, and doctor-likeness respectively. For GPT-2, the "large" version is used.

learning approach, which uses a masked-token prediction (Devlin et al., 2018) task to regularize the dialog generation task. Given the conversation histories in the training set, we encode them using an encoder. Then on top of the encodings, two tasks are defined. One is the dialog generation task, which takes the encoding of a conversation history as input and predicts its corresponding response. The prediction is conducted using a dialog decoder. The other task is masked-token prediction (MTP). In MTP, some percentage of the input tokens are masked at random. The text with masked tokens is fed into the text encoder which learns a latent representation for each token including the masked ones. The task is to predict these masked tokens by feeding the final hidden vectors (produced by the encoder) of the masked tokens into an output softmax operation over the vocabulary. The loss of the MTP task serves as a data-dependent regularizer of the encoder to prevent the encoder from overfitting to the data-deficient dialog generation task. Formally, the method solves the following optimization problem:

$$\mathcal{L}^{(g)}(H, R; W^{(e)}, W^{(g)}) + \lambda \mathcal{L}^{(p)}(H; W^{(e)}, W^{(p)})$$

where H represents the conversation histories and R represents their corresponding responses. $W^{(e)}$, $W^{(g)}$, and $W^{(p)}$ denote the encoder, decoder in the dialog generation task, and prediction head in the MTP task respectively. $\mathcal{L}^{(g)}$ denotes the generation loss and $\mathcal{L}^{(p)}$ denotes the MTP loss. λ is a tradeoff parameter.

5 Experiments

We compare with the following baselines: Transformer (Vaswani et al., 2017), GPT-2 (Radford et al., b), unregularized BART (Liu et al., 2019), unregularized BERT-GPT (Wu et al., 2019), and task adaptive pretraining (TAPT) (Gururangan et al., 2020).

5.1 Experiments on the English Dataset

5.1.1 Experimental Settings

For the English dataset, we split it into a training, a validation, and a test set based on dialogs, with a ratio of 8:1:1. Table 2 shows the statistics of the data split. The hyperparameters were tuned on the validation dataset. Our method and TAPT are both applied to the BART encoder, where the probability for masking tokens is 0.15. If a token t is chosen to be masked, 80% of the time, we replace t with a special token `[MASK]`; 10% of the time, we replace t with a random word; and for the rest 10% of the time, we keep t unchanged. For the regularization parameter λ, we set it to 0.1.

We perform human evaluation of the generated responses. Five medical students are asked to give ratings (from 1 to 5, higher is better) to generated responses in four aspects: 1) Correctness: whether the response is clinically correct; 2) Relevance: how relevant the response is to the conversation history; 3) Informativeness: how much medical information and suggestions are given in the response; and 4) Doctor-like: how the response sounds like a real doctor. The responses are de-identified: annotators do not know which method a response is generated by. The groundtruth response from the doctor is also given ratings (in an anonymous way). Human evaluation was conducted on the test examples in the CovidDialog-English dataset. The ratings from different annotators are averaged.

We also performed automatic evaluation, using metrics including perplexity, NIST-n (Doddington, 2002) (where $n = 4$), BLEU-n (Papineni et al., 2002) (where $n = 2$ and 4), METEOR (Lavie and Agarwal, 2007), Entropy-n (Zhang et al., 2018) (where $n = 4$), and Dist-n (Li et al., 2015) (where $n = 1$ and 2).

5.1.2 Results on the English Dataset

Table 3 shows the human evaluation results. From this table, we make the following observations. **First**, our method outperforms the unregularized BART on all metrics. This demonstrates the effectiveness of our method in alleviating overfitting

	Transformer	GPT-2			BART	BART+TAPT	Ours
		Small	Medium	Large			
Perplexity	263.1	28.3	17.5	18.9	15.3	15.0	**14.9**
NIST-4	0.71	1.90	2.01	**2.29**	1.88	1.89	2.04
BLEU-2	7.3%	9.6%	9.4%	**11.5%**	8.9%	7.7%	8.3%
BLEU-4	5.2%	6.1%	6.0%	**7.6%**	6.0%	3.4%	5.0%
METEOR	5.6%	9.0%	9.5%	**11.0%**	10.3%	9.2%	9.8%
Entropy-4	5.0	6.0	**6.6**	**6.6**	6.5	6.3	**6.6**
Dist-1	3.7%	9.5%	16.6%	13.9%	**16.8%**	11.5%	15.7%
Dist-2	6.4%	22.9%	**36.7%**	31.0%	35.7%	25.3%	33.1%
Avg. Len	40.0	51.3	50.1	54.4	45.4	59.8	50.8

Table 4: Performance on the CovidDialog-English test set.

Patient: i have all the symptoms except fever, i went to medicross and dr said i can get tested if i want to i'm not sure if i should. she gave me antibiotics klacid xl 500mg, she said i can take it if i feel worse i'm worried it will make immune system bad?
Groundtruth response: in brief: antibiotic i don't recommend antibiotics for a simple viral upper respiratory tract infection unless examination revealed signs of acute bronchitis or sinusitis. they are not effective for viral infections like covid 19 with no bacterial lung involvement either. if you've been exposed to someone with covid 19 or or if you or someone you were exposed to travelled to a region where it was endemic, get tested would you like to video or text chat with me?
Transformer: hello and welcome to 'ask a doctor' service.i have reviewed your query and here is my advice. i have reviewed your query. let me know if i can assist you further.
DialoGPT-large: in brief: do you have access to a telephonic consult, or do you need to go in person? would you like to video or text chat with me? i can understand your concern. i have reviewed your query and here is my advice. yes, telephonically would be best. would you be happy to chat with you? i have answered your query. let me know if i can assist you further. regards, dr jnikolla, pulmonologist
BART: yes, you should get tested for covid-19. get tested if you have fever, cough or shortness of breath. if you are a smoker or have been in contact with someone with covid, get tested. would you like to video or text chat with me?
BART+TAPT: if you have been in contact with someone who has been diagnosed with covid-19 then you should definitely get tested. it will make your immune system stronger. hope i have answered your query. let me know if i can assist you further. take care regards, dr. jnikolla
Ours: you do not need to be tested for covid-19 at this time. if you develop fever, cough, shortness of breath and/or difficulty breathing, then you need to consult your pcp and get tested.

Table 5: Generated responses on a test example in the CovidDialog-English dataset.

and improving generalization performance. Our method encourages the encoder to solve an additional MTP task, which reduces the risk of overfitting to the data-deficient response generation task on the small-sized training dialogs. **Second**, our method performs better than BART+TAPT. Our method and TAPT both leverage conversation histories for masked-token prediction. The difference is: TAPT uses these histories to pretrain the encoder while our method uses these histories to regularize the encoder during finetuning. In our method, the encoder is learned to perform the response generation task and MTP task simultaneously. Thus the encoder is not completely biased to the generation task. In TAPT, the encoder is first learned by performing the MTP task, then finetuned by performing the generation task. There is a risk that after finetuning, the encoder is largely biased to the generation task on the small-sized training data, which leads to overfitting. **Third**, our method achieves a doctor-like score that is close to the groundtruth. This indicates that the responses generated by our method have high language quality. The relevance rating of our method is higher than 3, which indicates a good level of relevance between the generated responses and conversation histories. The informativeness rating of our method is better than baselines, but still has a large gap with that of the groundtruth. Additional efforts are needed to improve informativeness, such as incorporating medical knowledge.

Table 4 summarizes the automatic evaluation results achieved by different methods. From this table, we make the following observations. **First**, our method achieves lower (better) perplexity (which is a relatively more reliable metric among various automatic metrics) than the baselines, which further demonstrates the effectiveness of our approach. **Second**, on machine translation metrics including NIST-4, BLEU-2, BLEU-4, and METEOR, the GPT2-large model achieves the highest scores. However, as noted in (Liu et al., 2016), machine translation metrics are not very reliable for evaluating dialog systems. **Third**, on diversity metrics including Entropy-4, Dist-1, and Dist-2, the GPT2-Medium model performs better than other methods. The average length of the generated responses by different methods is close to that of the groundtruth, which is around 50.

Table 5 shows an example of generating a doctor's response given the utterance of a patient. As can be seen, the response generated by our method is more relevant, informative, and human-like, compared with those generated by other baselines. It

	Transformer	GPT-2		BERT-GPT	BERT-GPT-TAPT	Ours
		No MMI	MMI			
Perplexity	53.3	22.1	25.7	10.8	9.3	**9.0**
NIST-4	0.39	0.43	**0.46**	0.36	0.30	0.37
BLEU-2	5.7%	6.2%	**7.2%**	4.6%	5.1%	5.4%
BLEU-4	4.0%	4.0%	**5.4%**	2.8%	2.6%	3.9%
METEOR	13.5%	13.9%	**14.3%**	12.2%	11.9%	13.0%
Entropy-4	7.9	9.0	**9.1**	8.5	7.8	7.9
Dist-1	5.5%	5.9%	3.2%	7.9%	**9.1%**	7.1%
Dist-2	29.0%	38.7%	35.7%	39.5%	**39.7%**	36.6%
Avg Len	19.3	35.0	58.7	21.6	13.9	20.6

Table 6: Performance on the CovidDialog-Chinese test set.

Split	#dialogs	#utterances	#pairs
Train	870	7844	3922
Validation	109	734	367
Test	109	916	458

Table 7: Chinese dataset split statistics

	C	R	I	D
Transformer	1.94	2.09	2.03	2.61
GPT-2	1.72	1.87	1.69	1.78
BERT-GPT	2.15	2.70	2.32	3.02
TAPT	2.27	2.68	2.42	3.11
Ours	**2.87**	**2.77**	**2.49**	**3.19**
Groundtruth	3.11	3.47	3.22	3.71

Table 8: Human evaluation on CovidDialog-Chinese test set. C, R, I, and D represent correctness, relevance, informativeness, and doctor-likeness respectively. Our method and TAPT are based on BERT-GPT. In GPT-2, no maximum mutual information (MMI) is used.

gives correct and informative medical advice such as "if you develop fever, cough, shortness of breath and/or difficulty breathing, then you need to consult your pcp and get tested" and has correct grammar and semantics. In contrast, BART gives clinically incorrect responses such as if someone is a smoker, he or she should be tested for COVID-19. So does BART+TAPT, which incorrectly suggests that getting tested will make the immune system stronger. The responses from GPT2-large and Transformer do not contain any useful medical advice.

5.2 Experiments on the Chinese Dataset

Based on dialogs, we split the Chinese dataset into a training set, validation set, and test set, with a ratio of 8:1:1. Table 7 shows the statistics of the data split. The regularization parameter λ was set to 0.8. Human evaluation was conducted by 5 medical students, on 100 randomly-sampled examples from the test set of CovidDialog-Chinese. The ratings from different annotators are averaged.

5.2.1 Results on the Chinese Dataset

Table 8 shows the human evaluation results. Our method and TAPT are based on BERT-GPT. As can be seen, our approach outperforms unregularized BERT-GPT. This further demonstrates the effectiveness of our approach in alleviating overfitting and improving generalization performance. In addition, our method outperforms TAPT. This further demonstrates that it is more beneficial to perform MTP and dialog generation jointly than separately.

Table 6 summarizes the automatic evaluation results. Our method achieves the lowest (best) perplexity among all methods. Our method outperforms unregularized BERT-GPT and BERT-GPT-TAPT on machine translation metrics as well. GPT2-MMI achieves the highest scores on machine translation metrics. BERT-GPT-TAPT performs better than other methods on diversity metrics.

6 Conclusions

In this paper, we make the first attempt to develop dialog generation models about COVID-19. We first collected two datasets – CovidDialogs – which contain medical conversations between patients and doctors about COVID-19. To alleviate the risk of overfitting, we develop a multi-task learning approach, which uses a masked-token prediction task to regularize the dialog generation model. Human evaluation and automatic evaluation results demonstrate the effectiveness of our proposed method in alleviating overfitting and generating clinically meaningful and linguistically high-quality dialogs about COVID-19.

Acknowledgement

This work was supported by gift funds from Tencent AI Lab and Amazon AWS.

Broader Impact

Dialog systems developed using the collected data in this work should be used very cautiously, under the guidance and supervision of physicians and with approval from the Food and Drug Administration. These dialog systems have the potential to provide timely and accessible COVID-19 consultations to the general public, especially to those who are underserved medically. However, clinical consultation is mission-critical. If the dialog systems make clinical errors, they may cause negative health issues to users. Therefore, these dialog systems should be used as assistants to physicians, rather than operating independently without human supervision. The collected dialogs are from public medical forums, which may be largely different from the patient-doctor dialogue in clinics and hospitals. Such a bias should be paid attention to when using this dataset.

References

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. 2017. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): a randomized controlled trial. *JMIR mental health*, 4(2):e19.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

David Ireland, Christina Atay, Jacki Liddle, Dana Bradford, Helen Lee, Olivia Rushin, Thomas Mullins, Dan Angus, Janet Wiles, Simon McBride, et al. 2016. Hello harlie: enabling speech monitoring through chat-bot conversations. In *Digital Health Innovation for Consumers, Clinicians, Connectivity and Community-Selected Papers from the 24th Australian National Health Informatics Conference, HIC 2016, Melbourne, Australia, July 2016.*, volume 227, pages 55–60. IOS Press Ebooks.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Liliana Laranjo, Adam G Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie YS Lau, et al. 2018. Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association*, 25(9):1248–1258.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.

I. Loshchilov and F. Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Gale M Lucas, Albert Rizzo, Jonathan Gratch, Stefan Scherer, Giota Stratou, Jill Boberg, and Louis-Philippe Morency. 2017. Reporting mental health symptoms: breaking down barriers to care with virtual human interviewers. *Frontiers in Robotics and AI*, 4:51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of*

the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics.

Pierre Philip, Jean-Arthur Micoulaud-Franchi, Patricia Sagaspe, Etienne De Sevin, Jérôme Olive, Stéphanie Bioulac, and Alain Sauteraud. 2017. Virtual human as a new diagnostic tool, a proof of concept study in the field of major depressive disorders. *Scientific reports*, 7(1):1–7.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. a. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. b. Language models are unsupervised multitask learners.

Hyekyun Rhee, James Allen, Jennifer Mammen, and Mary Swift. 2014. Mobile phone-based asthma self-management aid for adolescents (masmaa): a feasibility study. *Patient preference and adherence*, 8:63.

Hiroki Tanaka, Hideki Negoro, Hidemi Iwasaka, and Satoshi Nakamura. 2017. Embodied conversational agents for multimodal automated social skills training in people with autism spectrum disorders. *PloS one*, 12(8).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuan-Jing Huang, Kam-Fai Wong, and Xiang Dai. 2018. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–207.

Qingyang Wu, Lei Li, Hao Zhou, Ying Zeng, and Zhou Yu. 2019. Importance-aware learning for neural headline editing. *arXiv preprint arXiv:1912.01114*.

Yuan Xia, Jingbo Zhou, Zhenhui Shi, Chao Lu, and Haifeng Huang. Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis.

Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7346–7353.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, pages 1810–1820.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Appendix

A Related Works

A.1 Medical Dialog Generation

Many works have been devoted to developing medical dialog systems. Please refer to (Laranjo et al., 2018) for a comprehensive review. Some methods (Lucas et al., 2017; Philip et al., 2017; Tanaka et al., 2017) predefine a sequence of steps or states which are used to guide the conversation. Other methods (Rhee et al., 2014; Ireland et al., 2016; Fitzpatrick et al., 2017) use predetermined templates to extract information from the conversation history and use rules to generate responses from the filled slots in the templates. These methods rely heavily on knowledge engineering and are difficult to be quickly adapted to a new and time-sensitive task such as COVID-19 dialog generation.

A.2 Self-supervised Learning for NLP

Self-supervised learning (SSL) aims to learn meaningful representations of input data without using human annotations. It creates auxiliary tasks solely using the input data and forces deep networks to learn highly-effective latent features by solving these auxiliary tasks. In NLP, various auxiliary tasks have been proposed for SSL, such as next token prediction in GPT (Radford et al., a), masked token prediction in BERT (Devlin et al., 2018), text denoising in BART (Lewis et al., 2019), and so on. These models have achieved substantial success in learning language representations. The GPT model (Radford et al., a) is a language model (LM) based on Transformer (Vaswani et al., 2017). Unlike Transformer which defines a conditional probability on an output sequence given an input sequence, GPT defines a marginal probability on a single sequence. In GPT, the conditional probability of the next token given the historical sequence is defined using the Transformer decoder. The weight parameters are learned by maximizing the likelihood on the sequence of tokens. BERT (Devlin et al., 2018) aims to learn a Transformer encoder for representing texts. BERT's model architecture is a multi-layer bidirectional Transformer encoder. In BERT, the Transformer uses bidirectional self-attention. To train the encoder, BERT masks some percentage of the input tokens at random, and then predicts those masked tokens by feeding the final hidden vectors (produced by the encoder) corresponding to the masked tokens into an output softmax over vocabulary. BERT-GPT (Wu et al., 2019) is a model used for sequence-to-sequence modeling where a pretrained BERT is used to encode the input text and GPT is used to generate the output text. In BERT-GPT, the pretraining of the BERT encoder and the GPT decoder is conducted separately, which may lead to inferior performance. Auto-Regressive Transformers (BART) (Lewis et al., 2019) has a similar architecture as BERT-GPT, but trains the BERT encoder and GPT decoder jointly. To pretrain the BART weights, the input text is corrupted randomly, such as token masking, token deletion, text infilling, etc., then the network is learned to reconstruct the original text. ALBERT (Lan et al., 2019) uses parameter-reduction methods to reduce the memory consumption and increase the training speed of BERT. It also introduces a self-supervised loss which models inter-sentence coherence.

B Datasets

Table 9 shows an example of the English dataset.

C Experiments

C.1 Baselines

We compare the following baselines.

- **Transformer**. A conversation history is fed into the Transformer (Vaswani et al., 2017) encoder and the encoding is fed into the Transformer decoder to generate the corresponding response. The weights of the encoder and decoder are initialized randomly.
- **GPT-2**. Given a dialog history s and a ground-truth response $t = x_1, \cdots, x_n$, a GPT-2 model (Radford et al., a) is trained to maximize the following probability: $p(t|s) = p(x_1|s) \prod_{i=2}^{n} p(x_i|s, x_1, \cdots, x_{i-1})$, where conditional probabilities are defined by the Transformer decoder. For experiments on CovidDialog-English, the GPT-2 model is pretrained on English Reddit dialogs (Zhang et al., 2019). For experiments on CovidDialog-Chinese, the GPT-2 model is pretrained on Chinese chatbot corpus[1].
- **Unregularized BART** (Liu et al., 2019). This approach is the same as Transformer, except that the encoder and decoder are initialized using the pretrained BART (Lewis et al., 2019). The encoder and decoder are finetuned on CovidDialog-

[1] https://github.com/codemayq/chinese_chatbot_corpus

Description of patient's medical condition: I have a little fever with no history of foreign travel or contact. What is the chance of Covid-19?
Dialog **Patient**: Hello doctor, I am suffering from coughing, throat infection from last week. At that time fever did not persist and also did not feel any chest pain. Two days later, I consulted with a doctor. He prescribed Cavidur 625, Montek LC, Ambrolite syrup and Betaline gargle solution. Since then throat infection improved and frequent cough also coming out. Coughing also improved remarkably though not completely. From yesterday onwards fever is occuring (maximum 100-degree Celcius). I have not come in touch with any foreign returned person nor went outside. In our state, there is no incidence of Covid-19. Please suggest what to do? **Doctor**: Hello, I can understand your concern. In my opinion, you should get done a chest x-ray and CBC (complete blood count). If both these are normal then no need to worry much. I hope this helps. **Patient**: Thank you doctor. After doing all these I can upload all for further query. **Doctor**: Hi, yes, upload in this query only. I will see and revert to you.

Table 9: An exemplary consultation in the CovidDialog-English dataset. It consists of a brief description of the patient's medical conditions and the conversation between the patient and a doctor.

	Transformer	BERT-GPT	Ours	TAPT	GPT-2
GPU	TITAN Xp	GeForce RTX 2080	GeForce GTX 1080Ti	GeForce GTX 1080Ti	TITAN Xp
Num. of GPUs	1	1	1	1	1
Runtime	105	230	488	240	27

Table 10: Computing infrastructure and runtime (seconds per epoch) on CovidDialog-Chinese

English. During finetuning, no self-supervised regularization is used.

- **Unregularized BERT-GPT.** This approach is the same as Transformer, except that the encoder is initialized using pretrained BERT and the decoder is initialized using pretrained GPT-2. BERT and GPT-2 are both pretrained on large-scale Chinese corpus (Cui et al., 2019). The encoder and decoder are finetuned on CovidDialog-Chinese. During finetuning, no self-supervised regularization is used.
- **Task adaptive pretraining (TAPT)** (Gururangan et al., 2020). In this approach, given the Transformer encoder pretrained using BART/BERT on large-scale external corpora, it is further pretrained by predicting masked tokens on the input conversation histories in the CovidDialog datasets (without using output responses). Then the encoder is finetuned by predicting the responses from conversation histories. Similar to our method, TAPT also performs masked-token prediction (MTP) on conversation histories. The difference is: TAPT performs the MTP task and the generation task sequentially while our method performs these two tasks jointly.

C.2 Experimental Settings

C.2.1 Experimental Settings on the English Dataset

For GPT-2, we used three variants (Zhang et al., 2019) with different sizes: small, medium, and large, with 117M, 345M, and 762M weight parameters respectively. Maximum mutual information was not used. We used the Adam (Kingma and Ba, 2014) optimizer for the Transformer model and the AdamW (Loshchilov and Hutter, 2017) optimizer for other models. For all methods except TAPT, we used the optimizer with linear learning rate scheduling, setting the initial learning rate as 4e-5 and the batch size as 4. We perform TAPT for 100 epochs, setting the initial learning rate as 1e-4 and the batch size as 256. The objective for dialog generation is the cross entropy loss with label smoothing where the factor was set to 0.1. For pretrained models, we finetune them on the CovidDialog-English dataset for 5 epochs, while for the un-pretrained Transformer, we train it for 50 epochs. We set a checkpoint at the end of every epoch and finally take the one with the lowest perplexity on validation set as the final model. In response generation, for all models, we use beam search with beam width of 10 during decoding.

Among the automatic evaluation metrics, BLEU, METEOR, and NIST are common metrics for evaluating machine translation. They compare the similarity between generated responses and the ground-truth by matching n-grams. NIST is a variant of BLEU, which weights n-gram matches using information gain to penalize uninformative n-grams. Perplexity is used to measure the quality and smoothness of generated responses. Entropy and Dist are used to measure lexical diversity of generated responses. For perplexity, the lower, the

	Transformer	BERT-GPT	Ours	TAPT	GPT-2
Num. of epochs	30	2	2	2	8
Validation loss	3.17	1.90	2.10	2.08	2.90
Validation perplexity	32.74	6.68	8.14	7.98	18.94

Table 11: Validation performance on CovidDialog-Chinese

Transformer	90M
BERT-GPT	203M
GPT-2	81M

Table 12: Number of weight parameters of each model on CovidDialog-Chinese

	GPU	Runtime
Transformer	GeForce GTX 1080 Ti $\times$ 4	72
GPT-2	GeForce GTX 1080 Ti $\times$ 4	252
BART	GeForce GTX 1080 Ti $\times$ 4	180
Ours	Tesla P100-PCIE-16GB $\times$ 1	270
TAPT	Tesla P100-PCIE-16GB $\times$ 1	150

Table 13: Computing infrastructure and runtime (seconds per epoch) on the CovidDialog-English dataset

better. For other metrics, the higher, the better. As noted in (Liu et al., 2016), while automatic evaluation is useful, they are not completely reliable. Among these metrics, perplexity is generally considered to be more reliable than others.

C.2.2 Experimental Settings on the Chinese Dataset

The hyperparameters were tuned on the validation set. We stop the training procedure when the validation loss stops to decrease. Our method and TAPT are both applied to the BERT encoder in BERT-GPT, where the probability of masking tokens is 0.15. The encoder and decoder structures in BERT-GPT are similar to those in BERT, which is a Transformer with 12 layers and the size of the hidden states is 768. Network weights are optimized with stochastic gradient descent with a learning rate of 1e-4. In the finetuning of BERT-GPT, the max length of the source sequence and target sequence was set to 400. During decoding for all methods, beam search with $k = 50$ was used.

For GPT-2, we used the DialoGPT-small (Zhang et al., 2019) architecture where the number of layers in the Transformer was set to 10. The context size was set to 300. The embedding size was set to 768. The number of heads in multi-head self-attention was set to 12. The epsilon parameter in layer normalization was set to 1e-5. Network weights were optimized with Adam, with an initial learning rate of 1.5e-4 and a batch size of 8. The Noam learning rate scheduler with 2000 warm-up steps was used. For Transformer, we used the HuggingFace implementation[2] and followed their default hyperparameter settings. We evaluated the models using perplexity, NIST-4, BLEU-2, 4, METEOR, Entropy-4, and Dist-1, 2.

C.2.3 Additional Details about Human Evaluation

In human evaluation on CovidDialog-Chinese, we randomly select 100 examples. Each example includes a conversation history, groundtruth response, and responses generated by different methods. When presented to annotators, the groundtruth and responses generated by different methods are de-identified (given a response, annotators do not know which method generated this response) and randomly shuffled for different examples. The ratings from different annotators are averaged. In human evaluation on CovidDialog-English, we perform evaluation on all test examples.

C.3 Additional Analysis on Experimental Results

Additional analysis of results in Table 3 1) Pretrained models including GPT-2 and BART perform better than Transformer. This further demonstrates the effectiveness of pretraining. 2) BART performs better than GPT-2, though GPT-2 achieves better scores on machine translation metrics. This is in accordance with the results in (Liu et al., 2016) that machine translation metrics are not good for evaluating dialogue generation.

Additional analysis of results in Table 4 1) Pretrained models including GPT-2 and BART in general perform better than un-pretrained Transformer. This demonstrates the effectiveness of transfer learning, which leverages external large-scale data to learn powerful representations of texts. 2) BART achieves lower perplexity than GPT-2

[2]https://github.com/huggingface/transformers

	Transformer	GPT-2	TAPT	BART	Ours
Num. of epochs	100	5	5	5	5
Validation loss	8.02	3.06	2.88	2.84	2.87
Validation perplexity	260.30	21.50	17.74	17.28	17.56

Table 14: Validation performance on CovidDialog-English

Transformer	36M
GPT-2	768M
BART	406M

Table 15: Number of weight parameters of each model on CovidDialog-English

models. This is probably because BART is pretrained on a much larger and more diverse corpus than GPT-2, which enables BART to better model the language. 3) GPT2-large performs better than BART on machine translation metrics including NIST, BLEU, and METEOR. This is probably because GPT2-large is pretrained on dialogue data and therefore tends to generate n-grams that are more related to dialogues. 4) On diversity-related metrics including Entropy and Dist, BART is on par with GPT-2 models.

Additional analysis of results in Table 8 1) Pretrained BERT-GPT works better than unpretrained Transformer. Though pretrained, GPT-2 is not as good as Transformer. The possible reason is the training corpora of GPT-2 is daily dialogues, which has a large domain shift from medical dialogues. The performance gap between BERT-GPT and Groundtruth is larger than that between BART and Groundtruth, despite the number of Chinese training dialogues is larger than that of English training dialogues. This indicates that it is more challenging to develop COVID-19 dialogue systems on Chinese. One major reason is the Chinese dialogues are more noisy than the English ones, with a lot of incorrect grammar, abbreviations, semantic ambiguities, etc.

Additional analysis of results in Table 6 1) Pretrained models including GPT-2 and BERT-GPT achieve lower perplexity than Transformer. This further demonstrates the effectiveness of transfer learning. 2) GPT2-MMI achieves better scores than other methods on machine translation metrics, which is consistent with the results on the CovidDialog-English dataset. 3) BERT-GPT-TAPT achieves better Dist scores than other methods. We manually checked the generated responses by BERT-GPT-TAPT. Indeed, they are more diverse than others. 4) Maximum mutual information (MMI) does not have a clear efficacy in improving the quality of generated responses.

D Computing infrastructure, runtime, validation performance, number of weight parameters, implementation details

D.1 On Chinese CovidDialog

The computing infrastructure and runtime (seconds per epoch) on CovidDialog-Chinese is shown in Table 10. The validation performance is shown in Table 11. The number of weight parameters of each model on CovidDialog-Chinese is shown in Table 12.

We use PyTorch to implement all models. The version of Torch is 1.4.0 (or above). The python package "Transformers[3]" is 2.1.1 for GPT-2 and 2.8.0 (or above) for Transformer and BERT-GPT. When testing, we calculate NIST-n (Doddington, 2002), BLEU-n (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007) using NLTK[4] with version 3.5, and calculate Entropy-n (Zhang et al., 2018) and Dist-n (Li et al., 2015) based on the scripts in DialoGPT[5]. We use Gradient Accumulation in PyTorch to enlarge the mini-batch size to 32. Gradient Accumulation is a mechanism of PyTorch, which splits a large batch into smaller batches. The computation on smaller batches is executed sequentially. We set the number of gradient accumulation as 4 so that the mini-batch size is $8 * 4 = 32$.

D.2 On English CovidDialog

Table 13 shows the computing infrastructure and runtime (seconds per epoch) on the CovidDialog-English dataset. The number of epochs and validation performance of each model on CovidDialog-English are shown in Table 14. The number of weight parameters of each model on CovidDialog-English is shown in Table 15.

[3]https://github.com/huggingface/transformers
[4]https://www.nltk.org/
[5]https://github.com/microsoft/DialoGPT

Constructing Multi-Modal Dialogue Dataset by Replacing Text with Semantically Relevant Images

Nyoungwoo Lee*, **Suwon Shin***, **Jaegul Choo**, **Ho-Jin Choi**, and **Sung-Hyon Myaeng**

KAIST, Daejeon, South Korea

{*leenw2, ssw0093, jchoo, hojinc, myaeng*}*@kaist.ac.kr*

Abstract

In multi-modal dialogue systems, it is important to allow the use of images as part of a multi-turn conversation. Training such dialogue systems generally requires a large-scale dataset consisting of multi-turn dialogues that involve images, but such datasets rarely exist. In response, this paper proposes a 45k multi-modal dialogue dataset created with minimal human intervention. Our method to create such a dataset consists of (1) preparing and pre-processing text dialogue datasets, (2) creating image-mixed dialogues by using a text-to-image replacement technique, and (3) employing a contextual-similarity-based filtering step to ensure the contextual coherence of the dataset. To evaluate the validity of our dataset, we devise a simple retrieval model for dialogue sentence prediction tasks. Automatic metrics and human evaluation results on such tasks show that our dataset can be effectively used as training data for multi-modal dialogue systems which require an understanding of images and text in a context-aware manner. Our dataset and generation code is available at https://github.com/shh1574/multi-modal-dialogue-dataset.

1 Introduction

Humans often use images in instant messaging services to express their meaning and intent in the dialogue context. For a dialogue system such as a chatbot to respond to human users adequately in this kind of multi-modal situations, it is necessary to understand both images and texts in their context and incorporate them in the dialogue generation process.

Training such a multi-modal dialogue system generally requires a large amount of training data involving images and texts in various contexts. However, numerous existing approaches relying on image captioning (Lin et al., 2014; Young et al., 2014) or visual question answering (Mostafazadeh et al., 2016; Das et al., 2017) techniques had to be trained with the datasets mostly irrelevant to the dialogue context. In other words, images were interpreted independently of the dialogue context, due to the lack of sufficient multi-modal dialogue datasets.

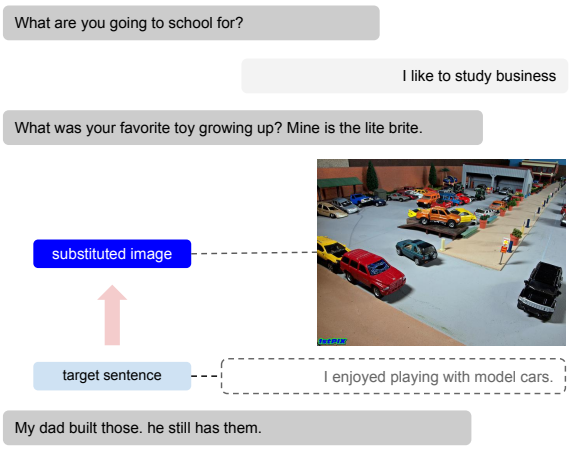


Figure 1: Example of multi-modal dialogue dataset.

Those datasets containing image-grounded conversations (Mostafazadeh et al., 2017; Shuster et al., 2020a) do not even cover the situations related to dialogue context before the image, because all conversations in the dataset always start from the given image. Although the relationship between images and texts can be learned using image-grounded conversations (Lu et al., 2019; Chen et al., 2020; Tan and Bansal, 2019; Su et al., 2020; Li et al., 2019b), it cannot still learn the dependency between the dialogue context before and after the image.

In this paper, we propose a 45k multi-modal dialogue dataset in the form of Fig. 1. Each multi-modal dialogue instance consists of a textual response and a dialogue context with multiple text

* Equal contribution.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 897–906
August 1–6, 2021. ©2021 Association for Computational Linguistics

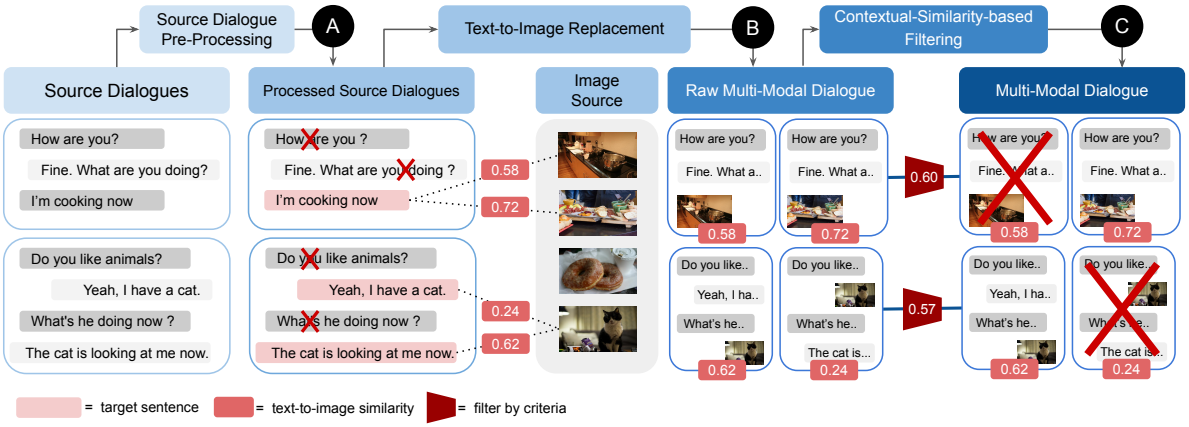


Figure 2: Overall pipeline for multi-modal dialogue dataset creation.

utterances and an image. To create this dataset, we start with existing text-only dialogue datasets as source dialogues, and then replace part of sentences in source dialogues with their semantically relevant images. The detailed steps include (1) source dialogue pre-processing, such as deleting a stop word, to improve the quality of similarity calculations, (2) creating dialogues containing an image by replacing a sentence with a similarity-based text-to-image replacement technique, and (3) pruning low-quality dialogues by employing a contextual-similarity-based filtering method. The overall process ensures that the created dataset consists of natural dialogue examples containing diverse images.

In order to validate our dataset creation process and examine the quality of our multi-modal dialogue dataset, we devise the task of predicting current and next dialogue sentences while considering the dialogue context and images. We also develop simple retrieval models to learn the relationship between images and texts for the tasks. Human evaluation results for predicting dialogue tasks show that the sentences are predicted as intended, i.e., in a context-aware manner, using the images. The results also show that our dataset can serve as practical training resources for multi-modal dialogue tasks that involve both image and dialogue context.

2 Multi-Modal Dialogue Generation

Our multi-modal dialogue dataset is constructed based on three source dialogue datasets and two image captioning datasets: DailyDialog (Li et al., 2017), EmpatheticDialogues (Rashkin et al., 2019), and Persona-Chat (Zhang et al., 2018) for the former and the MS-COCO (Lin et al., 2014) and Flicker 30k (Young et al., 2014) for the latter. The statistics of each dataset are summarized in Appendix A. After obtaining the source datasets, we replace sentences in the source dialogues with proper images by searching the image dataset to create image-mixed dialogues that maintain semantic coherence. To this end, we apply the three-stage method as shown in Fig. 2: (1) source dialogue pre-processing, (2) text-to-image replacement, and (3) contextual-similarity-based filtering.

Source Dialogue Pre-Processing We pre-process source dialogue datasets for the subsequent text-to-image replacement (A in Fig. 2). To select candidate dialogue sentences to be replaced by images, we first exclude the question sentences from the candidate dialogues because it is not realistic to infer back a question out of an image to put in the place of the question. This step filters out 25.08% of the total sentences in the source dialogue datasets. Second, we remove stop words from the source dialogue datasets, because they do not contain meaningful information. All the remaining sentences in dialogue contexts after the pre-processing step are considered as potential target sentences to replace.

Text-to-Image Replacement In this step, we create multi-modal dialogues containing images by replacing target sentences from the candidate dialogue sentences with appropriate images in the image dataset based on text-to-image similarity (B in Fig. 2). We calculate the similarity by the pre-trained Visual Semantic Reasoning Network (VSRN) (Li et al., 2019a), a state-of-the-art image-

text matching model based on text-to-image similarity. We first identify target sentences and then select candidate images for replacement using the threshold ensuring context coherence, as will be discussed in the subsequent contextual-similarity-based filtering step. Because we aim to maintain the comprehensive flow of the dialogue, we replace only one sentence with an image per dialogue. If multiple image candidates exist for a single sentence, we separate them into distinct image-mixed dialogue instances. In detail, such separated instances are all made up of the same dialogue context and text response except for substituted images.

Contextual-Similarity-based Filtering We employ a contextual-similarity-based filtering step to enhance the context coherence of the created image-mixed dialogues (C in Fig. 2). We filter out the dialogues where text-to-image similarity does not exceed the threshold determined by human annotators. For human annotators on the matching quality of an image, a total of 300 test dialogues are selected for each combination. Since we used three source dialogue datasets and two image datasets, we create six combinations of each dialogue dataset and each image dataset. Automatically created image-mixed dialogue instances are divided into ten segments based on the similarity values, and 30 are selected randomly from each. We hired a total of 18 annotators to evaluate 1,800 instances sampled from these six combinations. The evaluation system is described in Appendix C.

The human evaluation was conducted based on three questions for each instance:

- Q1: How well does the substituted image contain **key objects** in the target sentence?
- Q2: How well does the substituted image represent the **meaning** in the target sentence?
- Q3: When the image is substituted for the target sentence, how **consistent** is it with the **context** of the conversation?

Q1 and Q2 ask whether the substituted image contains the core meaning of the target sentence (on a 3-point scale). Q3 evaluates the context coherence of the created dialogue containing the image (on a 5-point scale). We assume that dialogues above the median of the evaluation score (2 for Q1, Q2, and 3 for Q3) are suitable for use as training instances. Based on this assumption, we determine the threshold for each combination by interpolating the median in the correlation graph of the evaluation results and the similarity (Appendix B). We then analyze the correlation between the score for each question and text-to-image similarity using Spearman's correlation analysis as shown in Table 1. Overall, the similarity values are positively correlated with the scores obtained for the questions. Since Q2 and Q3 are reasonably correlated with semantic similarity, the substituted images tend to reflect the meaning of the target and context sentences. Thus, the evaluation results indicate that the automatically created image-text pairs with high similarity can be used as multi-modal dialogues. We filter the generated multi-modal dialogues based on the determined similarities, and then set the filtered dialogues as our final dataset. The statistics of the final dataset are summarized in Table 2.

	Similarity	Q1	Q2	Q3
Similarity		**0.5893**	0.4422	0.4334
Q1			0.7103	0.6646
Q2				**0.7570**
Q3				

Table 1: Spearman's correlation ρ between three questions and text-to-image similarity.

	train	valid	test
# total dataset	**39956**	**2401**	**2673**
Avg length of dialogue turns	13.01	13.62	13.59
Avg length of sentences	51.47	50.76	50.70
# total unique images	12272	334	682
# total unique dialogues	13141	2148	2390
# total unique target sentences	21495	2400	2671
Avg # of substituted images in a dialogue	1.86	1.00	1.00
Avg # of targets in a dialogue	1.64	1.12	1.12

Table 2: Multi-modal dialogue dataset statistics for splits of training, validation, and test set.

Data Quality We evaluate the quality of our dataset to validate the proposed dataset creation method. To this end, we randomly sample 300 image-mixed dialogues from our final dataset. The evaluation proceeds in the same manner as before, but we add a new question Q4, which asks to choose the intent of the image used in the dialogue as one among (1) answering the question, (2) expressing emotional reactions, (3) proposing a new topic, and (4) giving additional explanations for the previous context. For Q1, Q2, and Q3, the average scores evaluated by three annotators are shown to be 2.56, 2.17, and 3.13, respectively, indicating that

Model	Task	R@1	R@5	Mean Rank
IR Baseline	Current	21.62	49.49	30.04
IR Baseline	Next	8.13	21.07	29.41
Retrieval Model	Current	**50.35**	**86.64**	**3.11**
Retrieval Model	Next	14.38	36.10	20.58

Table 3: Automatic evaluation results about retrieval models and an information retrieval baseline on the current and next dialogue prediction task.

the context of the conversation containing the substituted image is consistent in our dataset. For Q4, the responses from the annotators are distributed with 27.3%, 20.0%, 32.7%, and 14.7%, for the four intent types as mentioned above, indicating our dataset contains balanced intent types.

3 Experiments

3.1 Experimental Setup

We consider two dialogue sentence prediction tasks given an image and a dialogue: current dialogue prediction and next dialogue prediction for a given image. We use a simple retrieval model composed of three modules (Shuster et al., 2020a,b): Resnext-101 (Xie et al., 2017) for an image encoder, BERT (Devlin et al., 2019) for a text encoder, and the fusion module. As input for training the model, we use images and up to three dialogue sentences immediately before the images as dialogue context.

3.2 Automatic Evaluation

We perform quantitative comparisons that follow recent work (Shuster et al., 2020a) to find the optimal setting for our retrieval model (Appendix D). To evaluate the retrieval accuracy, we use the recall at 1 and 5 out of 100 candidates consisting of 99 candidates randomly chosen from the test set and 1 ground-truth sentence, called R@1/100 and R@5/100, respectively. We also use the mean reciprocal rank. We compare our model with a simple information retrieval baseline. The candidates of the baseline model are ranked according to their weighted word overlap between the target sentence and an image caption followed by dialogue context.

As shown in Table 3, the R@1 performance of the retrieval model obtained 50.35 and 14.38 on the current and next sentence prediction task, outperforming the baseline on both tasks. This result indicates that our dataset properly works as the training data to learn the relationship between images and dialogue context in dialogue sentence prediction tasks where images and dialogue context have to be considered together.

Model inputs	R@1	R@5	Mean Rank
Image Only	37.30	80.66	3.91
Dialogue Context Only	28.06	56.83	12.57
Image + Dialogue Context	**51.21**	**86.34**	**3.08**

Table 4: Ablation studies of our retrieval models on the current dialogue prediction task.

Model inputs	R@1	R@5	Mean Rank
Image Only	7.29	21.92	31.78
Dialogue Context Only	11.90	29.89	23.95
Image + Dialogue Context	**14.38**	**36.10**	**20.58**

Table 5: Ablation studies about our retrieval models on the next dialogue prediction task.

3.3 Ablation Study

We then conduct ablation studies by removing modalities (image and dialogue context) in turn to check whether unwanted correlations exist in our dataset. Since we created our training and test datasets by a semi-automatic data creation method, unwanted correlations can exist in datasets that can infer the correct answer without using the image and context simultaneously. Such correlations would prevent the model from properly learning the relationship between images and context.

As shown in Tables 4 and 5, the results first show that the recall measure for ground-truth answers in the model that considers both context and image is higher than the model considering only images. It indicates that the models in each task properly consider both images and dialogue context to predict sentences. To elaborate, the model that only considers images are likely to choose responses that do not match the dialogue context before the image. For example in a given dog photo shown during a sad mood conversation, the model that only considers images can generate an out-of-context response, such as "It is so cute.". On the other hand, in the same context, the model that considers both the context and the image could generate appropriate responses, such as "what is wrong with your dog?" or "I miss your dog.".

The overall tendency also shows that the model performance degrades when we delete each modality one by one. Such results suggest that our data creation process did not generate correlations that interfere with forming the relationship between images and dialogue context.

3.4 Human Evaluation

We create a new test set to confirm that the model can predict sentences well even on test dialogues that are not constructed in the same manner. To this end, two researchers manually created 100 multi-modal dialogues by adding images to source dialogues that were not used in our dataset generation process for human evaluation. We proceed with the evaluation with three annotators per each prediction task, using a question (on a 5-point scale) asking how much the sentences predicted by the model are relevant to the image and dialogue context. The average scores of three annotators for each task were shown to be 3.36 for the current turn prediction and 3.06 for the next turn prediction. The results indicate that the models can predict sentences in a context-aware manner even with dialogues organized by humans.

4 Conclusions

We present the multi-modal dialogue dataset consisting of 45k multi-turn dialogues containing semantically coherent images as well as the dataset creation method. Human evaluation results of our multi-modal dialogues reveal that context coherence is well maintained even if the sentence is replaced by an image, showing the validity of our dataset and data creation approach. We then evaluate our dataset using two multi-modal dialogue prediction tasks, demonstrating its effectiveness when training a dialogue system to learn the relationship between images and dialogue contexts. Our proposed data creation method can be applied when efficiently preparing large-scale multi-modal dialogue datasets that cover diverse multi-modal situations.

Acknowledgments

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2013-2-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services, No. 2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), and No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques)

References

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: learning universal image-text representations. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 104–120.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 326–335.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019a. Visual semantic reasoning for image-text matching. In *Proc. of the IEEE international conference on computer vision (ICCV)*, pages 4654–4662.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019b. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 986–995.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 740–755.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, pages 13–23.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 462–472.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1802–1813.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5370–5381.

Kurt Shuster, Samuel Humeau, Antoine Bordes, and Jason Weston. 2020a. Image-chat: Engaging grounded conversations. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2414–2429.

Kurt Shuster, Eric Michael Smith, Da Ju, and Jason Weston. 2020b. Multi-modal open-domain dialogue. *arXiv preprint arXiv:2010.01082*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: pre-training of generic visual-linguistic representations. In *Proc. the International Conference on Learning Representations (ICLR)*.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5100–5111.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1492–1500.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, pages 67–78.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2204–2213.

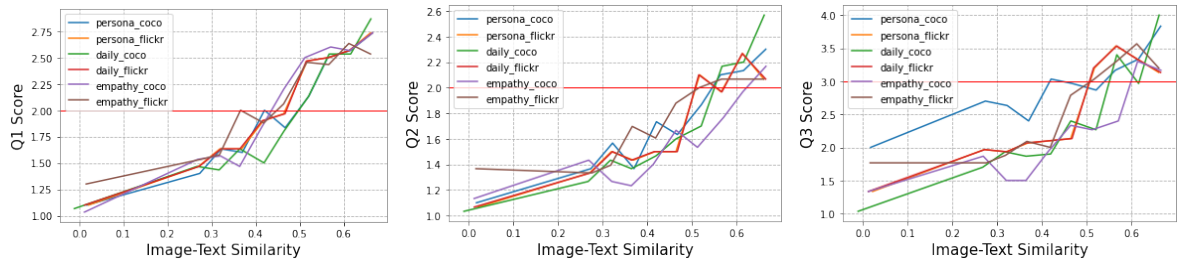


Figure 3: Correlation between text-to-image similarity and question scores (Q1, Q2, and Q3) for six combinations.

A Source Datasets Statistics

	type	training	validation	test
DailyDialogue	dialog	11118	1000	1000
Persona-Chat	dialog	8938	999	967
EmpatheticDialogues	dialog	17792	2758	2539
MS-COCO	image	113287	5000	5000
Flickr 30k	image	28000	1000	1000

Table 6: Source dialogue and image captioning dataset statistics for splits of training, validation, and test set.

B Detailed Description of Contextual-Similarity-based Filtering

	threshold	train	valid	test
Persona-COCO	0.546	11606	411	1136
Persona-Flickr	0.509	19148	1654	1014
Daily-COCO	0.555	3418	47	319
Daily-Flickr	0.619	141	6	5
Empathetic-COCO	0.623	245	2	11
Empathetic-Flickr	0.516	5398	281	188
Total		**39956**	**2401**	**2673**

Table 7: Number of data instances filtered by the thresholds for each combination

In this section, we analyze the human evaluation results for contextual-similarity-based filtering and determine thresholds for each dataset combination. The correlations between the similarity and evaluation results for each question are shown in Fig. 3. We assume that dialogue instances above the median of the evaluation score (2 for Q1, Q2, and 3 for Q3) are suitable for use in training. Based on the assumption, we determine the threshold for each combination by interpolating the median in the correlation graph of the evaluation results and the similarity. We select the largest one of three interpolated values of each question (Q1, Q2, and Q3). The data statistics for each combination filtered by the threshold are shown in Table 7.

Since the thresholds for each combination are determined differently, there are differences in the number of dialogue instances by combination. Such results suggest that the quality of multi-modal dialogue generation may vary depending on combining the text and image datasets. For example, the DailyDialog goes well with the MS-COCO but not with Flicker 30k. On the contrary, the EmpatheticDialogues goes well with the Flicker 30k but not with MS-COCO. Thus, we must consider finding the right combination among text and image datasets in the multi-modal dialogues generation process.

C Human Evaluation System

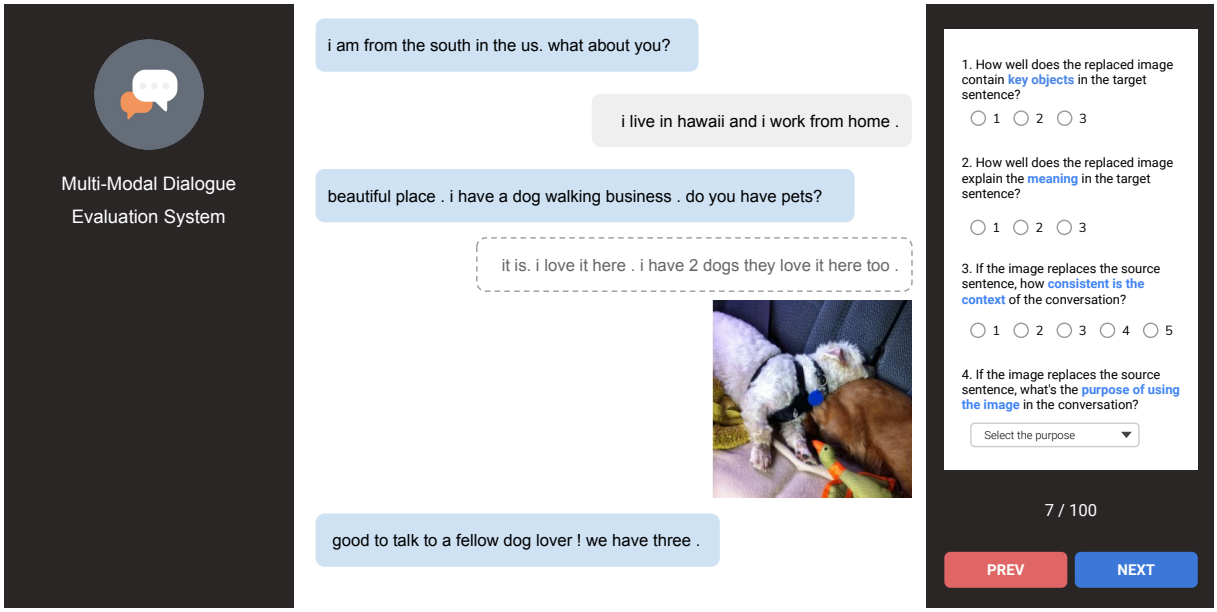


Figure 4: Human evaluation system for testing our multi-modal dialogue dataset.

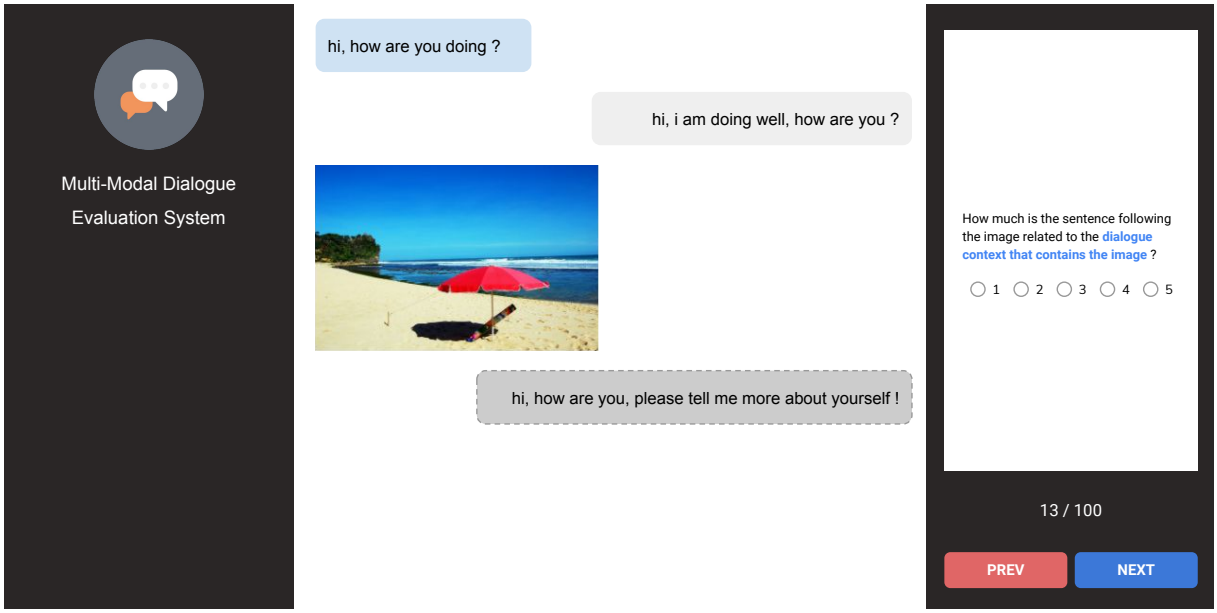


Figure 5: Human evaluation system for testing two dialogue sentence prediction tasks using our retrieval models.

In this section, we introduce the human evaluation system. We develop the system using a JavaScript library called ReactJS. Fig. 4 shows the implemented system for evaluating our multi-modal dialogue dataset. In this system, we ask users to evaluate a total of 100 dialog instances and answer three or four questions per instance. In addition to three questions described in Section 2, Q4[1] is added depending on the purpose of use. Fig. 5 shows the system for evaluating the performance of a retrieval model that performs dialog sentence prediction tasks. Similarly, we also ask users to evaluate a total of 100 dialog instances and answer one question per instance.

[1]Q4: If the image replaces the source sentence, what is the purpose of using the image in the conversation?

D Best Model Search

Model	Fusion Module	Image Encoder	Text Encoder	R@1	R@5	Mean Rank
$IRBaseline$	n/a	n/a	n/a	21.62	49.49	30.04
$RetrievalModel_{Att}$	Attention	Unfreeze	Freeze	11.74	39.13	15.73
$RetrievalModel_{Sum}$	Sum	Unfreeze	Freeze	9.95	35.13	15.73
$RetrievalModel_{Att}$	Attention	Unfreeze	Unfreeze	43.51	80.55	4.13
$RetrievalModel_{Sum}$	Sum	Unfreeze	Unfreeze	48.19	84.21	3.66
$RetrievalModel_{Att}$	Attention	Freeze	Unfreeze	48.41	85.97	3.40
$RetrievalModel_{Sum}$	Sum	Freeze	Unfreeze	**50.35**	**86.64**	**3.11**

Table 8: Comparison tests of the current dialogue prediction task on the multi-modal dialogue dataset. We compare different module variations and training strategies for our retrieval models.

Model	Fusion Module	Image Encoder	Text Encoder	R@1	R@5	Mean Rank
$IRBaseline$	n/a	n/a	n/a	8.13	21.07	29.41
$RetrievalModel_{Att}$	Attention	Unfreeze	Freeze	2.04	9.50	40.99
$RetrievalModel_{Sum}$	Sum	Unfreeze	Freeze	3.08	12.46	36.36
$RetrievalModel_{Att}$	Attention	Unfreeze	Unfreeze	4.09	15.95	32.07
$RetrievalModel_{Sum}$	Sum	Unfreeze	Unfreeze	13.38	33.93	21.10
$RetrievalModel_{Att}$	Attention	Freeze	Unfreeze	10.02	28.49	23.71
$RetrievalModel_{Sum}$	Sum	Freeze	Unfreeze	**14.38**	**36.10**	**20.58**

Table 9: Comparison tests of the next dialogue prediction task on the multi-modal dialogue dataset. We compare different module variations and training strategies for our retrieval models.

We compare different module options of our model. Each encoder has two options: whether to freeze or not during training, and the fusion module has two options: summation, and the attention-based transformer encoder. For final image-context fused representation, context and image representations are added in the summation fusion method, while two representations are concatenated, and then fed into the attention-based two-layer transformer encoder in the attention-based method. By this comparison, we decide to freeze only the image encoder and use the summation fusion method for both current and next dialogue prediction tasks.

We additionally show the results of an information retrieval baseline, which retrieves target dialogue using the tf-idf method between candidate dialogues and the caption of an image followed by dialogue context. As shown in Tables 8 and 9, our retrieval model significantly outperforms the information retrieval baseline, indicating that comprehensive understanding of context and images is helpful in multi-modal dialogues.

Our implementation uses an NVIDA TITAN RTX GPU for training, and training each epoch takes about 15 minutes. Our retrieval model using the summation fusion method has 204M parameters, while that using the attention-based fusion method has 254M parameters.

E Multi-Modal Dialogue Dataset Example

Figure 6: Our multi-modal dialogue dataset examples

For easy understanding of our dataset, we provide two additional examples of the multi-modal dialogue dataset in Fig. 6.

F Selected Example of Current Dialogue Prediction Task

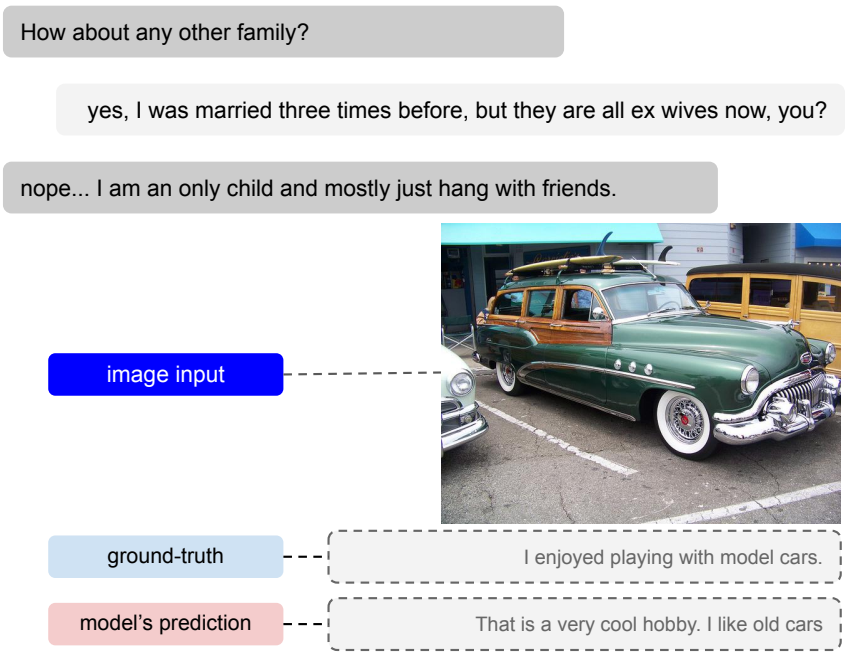


Figure 7: Ground-truth and dialogue sentence prediction example by our retrieval model used in the current turn prediction task.

Fig. 7 shows a reasonable example of a retrieved dialogue sentence by the retrieval model used in the current turn prediction task. Even if the model does not predict the ground-truth sentence, it can predict a plausible dialogue sentence.

Exposing the limits of Zero-shot Cross-lingual Hate Speech Detection

Debora Nozza
Bocconi University
Via Sarfatti 25, 20136
Milan, Italy
debora.nozza@unibocconi.it

Abstract

Reducing and counter-acting hate speech on Social Media is a significant concern. Most of the proposed automatic methods are conducted exclusively on English and very few consistently labeled, non-English resources have been proposed. Learning to detect hate speech on English and transferring to unseen languages seems an immediate solution. This work is the first to shed light on the limits of this zero-shot, cross-lingual transfer learning framework for hate speech detection. We use benchmark data sets in English, Italian, and Spanish to detect hate speech towards immigrants and women. Investigating post-hoc explanations of the model, we discover that non-hateful, language-specific taboo interjections are misinterpreted as signals of hate speech. Our findings demonstrate that zero-shot, cross-lingual models cannot be used as they are, but need to be carefully designed.

1 Introduction

An increasing propagation of hate speech has been detected on social media platforms (e.g., Twitter) where (pseudo-) anonymity enables people to target others without being recognized or easily traced. While this societal issue has attracted many studies in the NLP community, it comes with three important challenges. First, "hate speech" covers a **wide range of target types**, including misogyny, racism, and various other forms. While they often intersect, these types require different approaches.

Second, available labeled corpora refer to different definitions of hate speech, collection strategies, and annotation frameworks (Fortuna and Nunes, 2018). This lack of consistency strongly limits research on hate speech, which ultimately needs to apply cross-domain or transfer learning approaches for using different corpora.

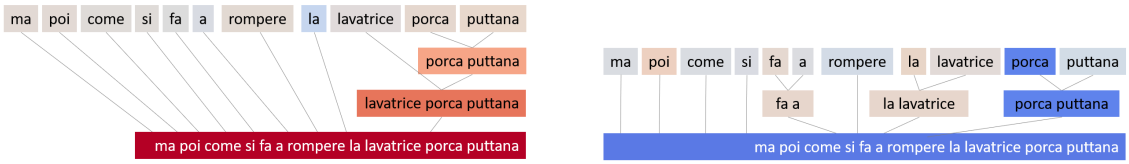
Third, most of the research on hate speech detection **consider only English** and only a **limited number of labeled corpora** are available (Fortuna and Nunes, 2018; Vidgen and Derczynski, 2021; Poletto et al., 2020). However, hate speech is not specific to any one language, and approaches proposed for English may not fit other languages. Each language exhibits different complexities in dealing with gender or reflecting cultural ideas around it.

The lack of models and labeled corpora for non-English languages seems a perfect application for zero-shot, cross-lingual learning (Lamprinidis et al., 2021; Bianchi et al., 2021). But is it? In this paper, we investigate the limitations of zero-shot, cross-lingual solutions based on mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) on benchmark data sets of hate speech against immigrants and women in English, Italian, and Spanish.

Our analysis demonstrates that these approaches have significant limitations: (1) they are not able to capture common (taboo) language-specific expressions, and (2) they do not transfer to different hate speech target types. We show that the reasons for these limitations are due to the high presence of language- and target-specific taboo interjections in non-hateful contexts, like *porca puttana* or *puta*.[1] **While derogatory for women, these terms are often used as intensifiers in non-hateful context, blurring the lines for detection**. Since English does not use equivalent words in the same way, zero-shot, cross-lingual models will not observe them in the training data. Consequently, these models consider the literal meaning of these terms as individual words, treating them as misogynous hate speech. These findings demonstrate that, at the current moment, cross-lingual, zero-shot transfer learning is not a solution for solving the lack of models and labeled corpora in non-English languages for hate speech detection.

[1] We report the uncensored words to ensure non-native speaker understanding.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 907–914
August 1–6, 2021. ©2021 Association for Computational Linguistics



(a) Misclassified prediction by zero-shot, cross-lingual model trained on English and Spanish and tested on Italian data.

(b) Correct prediction by monolingual model trained on Italian and tested on Italian data.

Figure 1: Hierarchical explanations of predictions of a non-hateful Italian tweet. Literal English translation: "how the hell can you break the washing machine".

Contributions 1) We investigate different learning frameworks on benchmark corpora for the detection of hate speech targeting women and immigrants 2) We expose the limits of zero-shot, cross-lingual solutions using the multilingual BERT model (mBERT) 3) We show interpretable results through post-hoc explanation.

2 Zero-shot, Cross-lingual Hate Speech Detection

We investigate different learning settings: 1) *zero-shot, cross-lingual*, i.e., training on one language and testing on unseen languages; 2) *monolingual*, i.e., training and testing on the same language; 3) *few-shot, cross-lingual*, i.e., training on one language and a small percentage of samples from the test language and testing on the test language; 4) *augmented cross-lingual*, i.e., training on several languages and testing on a language included in the training.

Multilingual BERT Recently, *contextual embeddings* pretrained on large corpora substantially advanced research for several major Natural Language Processing (NLP) tasks (Nozza et al., 2020). In particular, multilingual BERT (mBERT) (Devlin et al., 2019), a model pretrained on monolingual Wikipedia dumps in 104 languages, has shown surprisingly good abilities for zero-shot, cross-lingual model transfer for different NLP tasks (Pires et al., 2019). In this paper, we fine-tune the mBERT model on the task of hate speech detection considering data from one or multiple languages.

Post-hoc Explanation One of the biggest limitations of using complex black-box models, such as BERT, is the lack of interpretability. Following Kennedy et al. (2020), we use the Sampling and Occlusion (SOC) algorithm (Jin et al., 2020) to generate hierarchical explanations of predictions. SOC assigns an importance score to show how much a given word or sequence of words contributes to classifying a sentence as hate speech. Then, it combines this score hierarchically following semantic compositions. Visual representation examples are given in Figures 1 and 2. The hierarchy reflects how the model captures compositional semantics (e.g., stress or negation) in making predictions. Color intensity represents how much each phrase contributes to classifying the sentence as hate speech. The label prediction is encoded in the color: blue for non-misogynous and red for misogynous.

3 Data

	Immigrants			Women		
	EN	**IT**	**ES**	**EN**	**IT**	**ES**
Train	4500	2000	1618	4500	2500	2882
Dev	500	500	173	500	500	327
Test	1499	1000	800	1472	1000	799

Table 1: Corpora splits # of instances by target type.

To assess the cross-lingual evaluation framework, we use hate speech benchmark data sets with consistent definitions, annotation schema, and collection strategies (see Appendix C). For English and Spanish, we adopt the data sets proposed in the shared task of hate speech against immigrants and women on Twitter (HatEval) (Basile et al., 2019). For Italian, we consider two different corpora proposed for Evalita shared tasks (Caselli et al., 2018): the automatic misogyny identification challenge (AMI) (Fersini et al., 2018) for hate speech towards women, and the hate speech detection shared task on Facebook and Twitter (HaSpeeDe) (Bosco et al., 2018) for hate speech towards immigrants. Table 1 reports data distributions across languages and targets.

		Immigrants		
	Test	**IT**	**EN**	**ES**
Train	**IT**	*0.777*	*0.635***	*0.666*
	EN	*0.590***	*0.368*	*0.633*
	ES	*0.683***	*0.596***	0.630
	EN+ES	*0.706**	0.353	0.676*
	ES+IT	*0.757*	*0.538***	0.686*
	EN+IT	0.771	*0.340*	*0.657*
	Baseline	0.799	-	-

(a)

		Women		
	Test	**IT**	**EN**	**ES**
Train	**IT**	0.808	*0.545*	*0.463***
	EN	*0.449***	0.559	*0.546***
	ES	*0.337***	*0.558*	0.839
	EN+ES	*0.440*	0.449***	0.873*
	ES+IT	0.820	*0.502*	0.878*
	EN+IT	0.798	0.469***	*0.603***
	Baseline	0.844	-	-

(b)

Table 2: Macro-F1 results for the two hate speech targets. Monolingual results are underlined. Zero-shot cross-lingual results are highlighted in *italic*. * = differs significantly from monolingual at $p \leq 0.05$. ** = significant difference at $p \leq 0.01$.

4 Experimental Results

Table 2 shows the macro-averaged F1 score for hate speech detection on different training and test languages (in rows and columns, respectively). Underlined numbers refer to the monolingual setting results, while zero-shot, cross-lingual results are italicized. We report as *baselines* the best performing model for each of the considered data set released in conjunction with shared tasks.[2] Since the aim of this paper is to investigate classification abilities of cross-lingual, zero-shot models, we do not aim to overcome the baselines but to provide comparable results.

4.1 Hate speech towards immigrants

Observing monolingual results (underlined numbers in Table 2), we see that training and testing in English gives the poorest performance. This behavior is due to an over-sensitivity to specific words/hashtags used during data collection (e.g. *#SendThemBack*, *#StopTheInvasion*), which leads to overfitting. In Appendix A, we report the SOC explanation of a misclassified tweet containing these hashtags. We confirm this finding by training the monolingual English model on data deprived of these hashtags, which lead to higher macro-F1 (from 0.368 to 0.438).

The zero-shot, cross-lingual configuration (italic numbers in Table 2) shows very different results between the two targets. Zero-shot learning obtains good performance for detecting hate speech towards immigrants: when testing Italian and Spanish, results are very similar; when testing on English, training on a different language is better than including English data, resulting in a 22% macro-F1 improvement on average. This is because training sets based on other languages do not contain the above-mentioned specific words and therefore do not suffer from over-sensitization.

4.2 Hate speech towards women

Concerning hate speech towards women, *the zero-shot, cross-lingual model obtains significantly lower performance for Spanish and Italian*. To better understand this substantially different finding, we analyze wrongly labeled instances. We discover that zero-shot, cross-lingual models are strongly influenced by common, language-specific taboo interjections to mislabel non-hateful text as misogynous. In particular, expressions that contain literal insults towards women but are not misogynistic per se. For example in Spanish, beyond its misogynistic meaning, the word *puta* (literally *bitch*) is also used as an exclamation of surprise (e.g., *puta mierda*). The Italian expressions *porca troia* and *porca puttana* (literally *porca* (*pig*) + *troia*/*puttana* (*slut*)) are very generic taboo interjections that do not have a misogynistic connotation. It is important to notice that these interjections are not directly translatable and usually used in combination, e.g. *porca* + *puttana*, *puta* + *mierda*.

To demonstrate this finding, in Table 3 we report the number of times a zero-shot cross-learning model correctly predicts the labels of instances containing taboo interjections for Italian and Spanish (i.e., *porca puttana, porca troia, puta*). The high frequency of instances containing taboo interjections (29% and 78% of the test set), due also to the keyword-driven collection strategy, proves the importance of understanding these linguistic expressions. The following numbers illustrate the

[2] State-of-the-art performance do not exist for every combination, since Hateval (English and Spanish) consider hate speech towards women and immigrant in conjunction.

Test Lang	Frequency	Zero-Shot, Cross-Lingual	Monolingual
IT	294 (29%)	9 (3%)	291 (99%)
ES	627 (78%)	365 (58%)	514 (82%)

Table 3: Correct predictions for instances containing Italian and Spanish taboo interjections.

impact of taboo interjections: all the 276 Italian tweets containing *porca puttana* are labeled as non-misogynous and are consistently misclassified by zero-shot, cross-lingual model; the Spanish expression *hijo de puta* appears in 64 tweets (of which 57 are non-misogynous) for which the zero-shot, cross-lingual model achieves 62% accuracy vs. 90% accuracy of the monolingual model. We confirm this finding by training models on data deprived of these taboo interjections, obtaining improvements: 0.627 for **ES⇒IT**; 0.479 for **IT⇒ES**; 0.662 for **EN⇒IT**; 0.660 for **IT⇒EN**.

Figure 1 shows the SOC explanation of a non-hateful tweet correctly classified by the monolingual Italian model and wrongly classified by the zero-shot, cross-lingual model trained on English and Spanish data. As expected, training and testing on Italian teach the model that *porca puttana* is a very general exclamation that does not imply misogyny (high importance score for non-misogynous prediction). However, when training on other languages, this taboo interjection is not recognized because it is strictly related to the *test* language. We observe that zero-shot, cross-lingual models consider the literal meaning of individual words, and consequently treat terms like *porca puttana* as misogynous regardless of their use in context.

To further validate this major finding, we conduct an additional experiment on the corpus of hate speech towards women: we train *few-shot, cross-lingual* models randomly sampling 1% of training data in the test language. The averaged results on 10 runs in terms of macro-F1 are: 0.660 for **ES+EN⇒IT**; 0.702 for **EN+IT⇒ES**. The significant improvements with respect to zero-shot performances prove that misogyny detection is strongly entangled with common, language-specific taboo interjections that are very frequent in the data set.

4.3 Hate speech towards immigrants and women

Finally, to demonstrate the *need for treating target types separately*, we run the zero-shot, cross-lingual model on the merged data sets of hate speech towards immigrants and women. The results in terms of macro-F1 are: 0.572 for **ES+IT⇒EN**; 0.513 for **ES+EN⇒IT**; 0.632 for **EN+IT⇒ES** (see Appendix B).

Following Stappen et al. (2020), these scores suggest a sufficient adaptation by the models. However, they represent a compromise between the high results of zero-shot cross-lingual hate speech detection against immigrants and the low results of hate speech detection against women. By showing the results for the two separate targets, we demonstrated that zero-shot cross-lingual models suffer from limitations when predicting hate speech detection against women and that, in general, zero-shot cross-lingual hate speech detection has yet to be solved.

4.4 Impact of language-specific taboo interjections on XLM-R

In order to understand whether common, language-specific taboo interjections play a role in other language models, we conducted experiments with XLM-R (Conneau et al., 2020). XLM-R is a large cross-lingual language model based on RoBERTa (Liu et al., 2019), trained on 2.5TB of filtered CommonCrawl data, which significantly outperformed mBERT on a variety of cross-lingual benchmarks.

XLM-R achieves high macro-F1 scores in monolingual settings for detecting hate speech towards women in Italian and Spanish (0.806 for **IT⇒IT**; 0.859 for **ES⇒ES**). Similar to the previously presented findings, we observe a significant drop of 36% in macro-F1 when considering the zero-shot cross-lingual settings (0.604 for **EN⇒IT**; 0.511 for **ES⇒IT**; 0.404 for **IT⇒ES**; 0.658 for **EN⇒ES**). This drop in macro-F1 is more evident when considering the performance when training on Spanish and testing on Italian and vice versa. These results on XLM-R bring more evidence about the role that language-specific taboo interjections have in impacting the performance.

5 Related Work

Hate speech detection has attracted great interest in the NLP community. This has led to the proposal of automatic detection approaches based on machine learning (Indurthi et al., 2019; Nozza et al., 2019; Fersini et al., 2020a; Kennedy et al., 2020; D'Sa et al., 2020, inter alia) and the creation of benchmark data sets, usually distributed through

shared tasks (Waseem and Hovy, 2016; Davidson et al., 2017; Founta et al., 2018; Bosco et al., 2018; Kumar et al., 2018; Wiegand et al., 2018; Basile et al., 2019; Fersini et al., 2018; Zampieri et al., 2020; Fersini et al., 2020b, inter alia).

Only a few studies have investigated hate speech detection across different languages. Steimel et al. (2019) asked which factors affect multilingual settings for German and English, concluding that a shared classification algorithm is not conceivable due to lack of corpora comparability. In Sohn and Lee (2019), the authors proposed a multi-channel model exploiting multilingual BERT and language-specific BERT for Chinese, English, German, and Italian. Finally, Stappen et al. (2020) proposed a novel, attention-based classification block for performing zero- and few-shot, cross-lingual learning on the HatEval data set. While they state that transfer learning is effective for hate speech detection, we argue that there is a need to investigate hate speech targets separately since these models consistently fail misogyny classification.

6 Conclusion

We demonstrate that cross-lingual, zero-shot transfer learning, in its traditional settings, is not a feasible solution for solving the lack of models and labeled corpora for hate speech detection. We argue that hate speech is language specific, and NLP approaches to identifying hate speech must account for that specificity and the adoption of related techniques must be done with care (Bianchi and Hovy, 2021). We plan to expand this evaluation to other languages and to investigate a solution based on bias mitigation (Nozza et al., 2019; Kennedy et al., 2020) and on pragmatic role-aware models (Holgate et al., 2018; Pamungkas et al., 2020) to reduce the impact of this problem on classification. Future work will also focus on modeling language's social factors (Hovy and Spruit, 2016; Hovy, 2018; Hovy and Yang, 2021), such as speaker and receiver characteristics, and study their impact on hate speech detection classifiers.

Ethical Considerations

We are aware that the inherent (gender) biases of sentence and word embeddings are affecting the model's performance on detecting hate speech towards women (Bolukbasi et al., 2016; Sheng et al., 2019; Nangia et al., 2020; Nozza et al., 2021). We believe that this issue plays a role in the classification models. However, in this paper we extensively demonstrate that the presence of taboo interjections is one of the main hurdles that specifically hinder zero-shot, cross-lingual hate speech detection results.

Finally, we want to highlight that the presented findings are specifically related to the considered languages and data sets. Hopefully, our work will generate more conscious research about the use of hate speech detection models in zero-shot, cross-lingual frameworks.

Acknowledgments

This project has partially received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 949944, INTEGRATOR). Thanks to Dirk Hovy, Federico Bianchi, Salvatore Alessandro Casa, Tommaso Fornaciari, and Silvia Terragni for their invaluable feedback. Debora Nozza is a member of the Bocconi Institute for Data Science and Analytics (BIDSA) and the Data and Marketing Insights (DMI) unit.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Federico Bianchi and Dirk Hovy. 2021. On the gap between adoption and understanding in NLP. In *Findings of the Association for Computational Linguistics: ACL 2021*. Association for Computational Linguistics.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021. Cross-lingual contextualized topic models with zero-shot learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the EVALITA 2018 hate speech detection task. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018)*, volume 2263, pages 1–9, Turin, Italy. CEUR.org.

Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. 2018. EVALITA 2018: Overview of the 6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. In *Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018)*, Turin, Italy. CEUR.org.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashwin Geet D'Sa, Irina Illina, Dominique Fohr, Dietrich Klakow, and Dana Ruiter. 2020. Label propagation-based semi-supervised learning for hate speech classification. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 54–59, Online. Association for Computational Linguistics.

Elisabetta Fersini, Debora Nozza, and Giulia Boifava. 2020a. Profiling Italian misogynist: An empirical study. In *Proceedings of the Workshop on Resources and Techniques for User and Author Profiling in Abusive Language*, pages 9–13, Marseille, France. European Language Resources Association (ELRA).

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2018. Overview of the EVALITA 2018 task on automatic misogyny identification (AMI). *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2018)*, 12:59.

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2020b. AMI @ EVALITA2020: Automatic misogyny identification. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior.

Eric Holgate, Isabel Cachola, Daniel Preoţiuc-Pietro, and Junyi Jessy Li. 2018. Why swear? analyzing and inferring the intentions of vulgar expressions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4405–4414, Brussels, Belgium. Association for Computational Linguistics.

Dirk Hovy. 2018. The social and the neural network: How to make natural language processing about people again. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 42–49, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Dirk Hovy and Shannon L. Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, Berlin, Germany. Association for Computational Linguistics.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602, Online. Association for Computational Linguistics.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. 2019. FERMI at SemEval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, Online. Association for Computational Linguistics.

Ritesh Kumar, Atul Kr. Ojha, Marcos Zampieri, and Shervin Malmasi, editors. 2018. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Sotiris Lamprinidis, Federico Bianchi, Daniel Hardt, and Dirk Hovy. 2021. Universal joy a data set and results for classifying emotions across languages. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 62–75, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making sense of language-specific BERT models. *arXiv preprint arXiv:2003.02912*.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. HONEST: Measuring hurtful sentence completion in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online. Association for Computational Linguistics.

Debora Nozza, Claudia Volpetti, and Elisabetta Fersini. 2019. Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence*, WI '19, page 149–155, New York, NY, USA. Association for Computing Machinery.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2020. Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management*, 57(6):102360.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 1–47.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Hajung Sohn and Hyunju Lee. 2019. MC-BERT4HATE: hate speech detection using multi-channel BERT for different languages and translations. In *2019 International Conference on Data Mining Workshops, ICDM Workshops 2019, Beijing, China, November 8-11, 2019*, pages 551–559. IEEE.

Lukas Stappen, Fabian Brunn, and Björn W. Schuller. 2020. Cross-lingual zero- and few-shot hate speech detection utilising frozen transformer language models and AXEL. *CoRR*, abs/2004.13850.

Kenneth Steimel, Daniel Dakota, Yue Chen, and Sandra Kübler. 2019. Investigating multilingual abusive language detection: A cautionary tale. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1151–1160, Varna, Bulgaria. INCOMA Ltd.

Bertie Vidgen and Leon Derczynski. 2021. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *PLOS ONE*, 15(12):1–32.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

A Additional Post-Hoc Explanation

Figure 2 shows the hierarchically clustered explanations from SOC for an example of non-hateful speech wrongly classified as hateful by the monolingual English model. It is evident how the (incorrect) high score of the hashtag eclipses the influence of non-hateful words such as *days*, *kids*, and *school*.

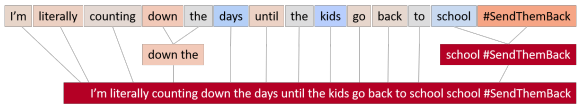


Figure 2: Hierarchical explanations of the incorrect prediction of a non-hateful English tweet by a monolingual model trained on English and tested on English data.

B Additional Results

		Immigrants+Women		
	Test	**IT**	**EN**	**ES**
Train	**IT**	0.804	*0.571***	*0.596***
	EN	*0.564***	0.416	*0.648***
	ES	*0.513***	*0.576***	0.752
	EN+ES	*0.513***	0.335***	0.768
	ES+IT	0.797	*0.572***	0.744
	EN+IT	0.802	0.399	*0.632***
	Baseline	-	0.651	0.730

Table 4: Results in terms of macro-F1 for the merged corpora containing hate speech towards immigrants and women. Monolingual results are underlined. Zero-shot cross-lingual results are highlighted in *italic*.
* = differs significantly from monolingual at $p \leq 0.05$.
** = significant difference at $p \leq 0.01$.

C Experimental Configuration

C.1 Consistent Data sets

We use benchmark hate speech data sets with consistent definitions, annotation schema, and collection strategies. All the three data sets (Bosco et al., 2018; Fersini et al., 2018; Basile et al., 2019) refer to the same definitions of hate speech towards immigrant and women.[3] This paper focuses on the common binary classification task (hateful/non-hateful) across all data sets, ensuring the same annotation schema. Finally, all data sets have been collected by following three strategies: (1) monitoring potential victims of hate accounts, (2) downloading the history of identified haters and (3) filtering Twitter streams with keywords, i.e. words, hashtags and stems.

For experimental evaluation, we use the data set splits provided in the associated shared task for comparability with previous work.

C.2 Implementation Details

We implement the proposed work exploiting the public code implementation of the classification model presented by Kennedy et al. (2020)[4]. We use their hyperparameter configuration for training: batch size is set to 32, the learning rate of the Adam optimizer is set to 2×10^{-5}, the loss function is the binary cross entropy.

Computing Infrastructure We independently run the experiments on two machines: the first one is equipped with two NVIDIA RTX 2080TI and has 64GB of RAM. The other one is equipped with four GPUs, NVIDIA GTX 1080TI, and has 32GB of RAM.

[3] https://github.com/msang/hateval/blob/master/annotation_guidelines.md

[4] https://github.com/BrendanKennedy/contextualizing-hate-speech-models-with-explanations

BERTTune: Fine-Tuning Neural Machine Translation with BERTScore

Inigo Jauregi Unanue[1,2], Jacob Parnell[1,2], Massimo Piccardi[1]
[1]University of Technology Sydney, NSW 2007, Australia
[2]RoZetta Technology, NSW 2000, Australia
Inigo.Jauregi@rozettatechnology.com
Jacob.Parnell@rozettatechnology.com
Massimo.Piccardi@uts.edu.au

Abstract

Neural machine translation models are often biased toward the limited translation references seen during training. To amend this form of overfitting, in this paper we propose fine-tuning the models with a novel training objective based on the recently-proposed BERTScore evaluation metric. BERTScore is a scoring function based on contextual embeddings that overcomes the typical limitations of n-gram-based metrics (e.g. synonyms, paraphrases), allowing translations that are different from the references, yet close in the contextual embedding space, to be treated as substantially correct. To be able to use BERTScore as a training objective, we propose three approaches for generating *soft predictions*, allowing the network to remain completely differentiable end-to-end. Experiments carried out over four, diverse language pairs have achieved improvements of up to 0.58 pp (3.28%) in BLEU score and up to 0.76 pp (0.98%) in BERTScore (F_{BERT}) when fine-tuning a strong baseline.

1 Introduction

Neural machine translation (NMT) has imposed itself as the most performing approach for automatic translation in a large variety of cases (Sutskever et al., 2014; Vaswani et al., 2017). However, NMT models suffer from well-known limitations such as overfitting and moderate generalization, particularly when the training data are limited (Koehn and Knowles, 2017). This mainly stems from the fact that NMT models have large capacity and are usually trained to maximize the likelihood of just a single reference sentence per source sentence, thus ignoring possible variations within the translation (e.g. synonyms, paraphrases) and potentially resulting in overfitting. A somewhat analogous problem affects evaluation, where metrics such as BLEU (Papineni et al., 2002) only consider as correct the predicted n-grams that match exactly in the ground-truth sentence. In order to alleviate the n-gram matching issue during evaluation, Zhang et al. (2020) have recently proposed the BERTScore metric that measures the accuracy of a translation model in a contextual embedding space. In BERTScore, a pretrained language model (e.g. BERT (Devlin et al., 2019)) is first used to compute the contextual embeddings of the predicted sentence, $\langle \hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_k \rangle$, and the reference sentence, $\langle \mathbf{y}_1, \ldots, \mathbf{y}_l \rangle$, with k and l word-pieces, respectively. Then, recall (R_{BERT}), precision (P_{BERT}), and F1 (F_{BERT}) scores are defined as cosine similarities between the normalized contextual embeddings. For example, the recall is defined as:

$$R_{BERT} = \frac{1}{|l|} \sum_{y_i \in y} \max_{\hat{y}_j \in \hat{y}} \mathbf{y}_i^T \hat{\mathbf{y}}_j \qquad (1)$$

where the $\max$ function acts as an alignment between each word in the reference sentence (y) and the words in the predicted sentence ($\hat{y}$). Conversely, P_{BERT} aligns each word of the predicted sentence with the words of the reference sentence, and F_{BERT} is the usual geometric mean of precision and recall. Note that with this scoring function a candidate and reference sentences with similar embeddings will be assigned a high score even if they differ completely in terms of categorical words. Zhang et al. (2020) have shown that this evaluation metric has very high correlation with the human judgment.

In this work, we propose using BERTScore as an objective function for model fine-tuning. Our rationale is that BERTScore is a sentence-level objective that may be able to refine the performance of NMT models trained with the conventional, token-level log-likelihood. However, in order to fine-tune the model with BERTScore as an objective, end-to-end differentiability needs to be ensured. While the BERTScore scoring function is based on word

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 915–924
August 1–6, 2021. ©2021 Association for Computational Linguistics

embeddings and is in itself differentiable, its input derives from categorical predictions (i.e. argmax or sampling), breaking the differentiability of the overall model. In this work, we solve this problem by generating *soft predictions* during training with three different approaches. One of the approaches, based on the Gumbel-Softmax (Jang et al., 2017), also leverages sampling, allowing the model to benefit from a certain degree of *exploration*. For immediacy, we refer to our approach as *BERTTune*. The experimental results over four, diverse language pairs have shown improvements of up to 0.58 pp (3.28%) in BLEU score and up to 0.76 pp (0.98%) in BERTScore with respect to a contemporary baseline (Ott et al., 2019).

2 Related Work

In recent years, various researchers have addressed the problem of overfitting in NMT models. This problem can be specially severe for neural models, given that, in principle, their large number of parameters could allow for a perfect memorization of the training set. For instance, Ma et al. (2018) have trained an NMT model using both a reference sentence and its bag-of-words vector as targets, assuming that the space of alternative, correct translations share similar bags-of-words. Others (Elbayad et al., 2018; Chousa et al., 2018) have proposed smoothing the probability distribution generated by the decoder using the embedding distance between the predicted and target words, forcing the network to increase the probability of words other than the reference. Another line of work has proposed to explicitly predict word embeddings, using the cosine similarity with the target embedding as the reward function (Kumar and Tsvetkov, 2019; Jauregi Unanue et al., 2019).

Reinforcement learning-style training has also been used to alleviate overfitting (Ranzato et al., 2016; Edunov et al., 2018). The use of beam search removes the exposure bias problem (Wiseman and Rush, 2016), and the use of sampling introduces some degree of exploration. In addition, these approaches allow using non-differentiable, sequence-level metrics as reward functions. However, in practice, approximating the expectation of the objective function with only one or a few samples results in models with high variance and convergence issues.

Significant effort has also been recently dedicated to leveraging large, pretrained language models (Devlin et al., 2019; Radford et al., 2018; Peters et al., 2018) for improving the performance of NMT models. This includes using contextual word embeddings either as input features (Edunov et al., 2019) or for input augmentation (Yang et al., 2020; Zhu et al., 2020), and using a pretrained language model for initializing the weights of the encoder (Clinchant et al., 2019). Alternatively, Baziotis et al. (2020) have proposed using a pretrained language model as a prior, encouraging the network to generate probability distributions that have a high likelihood in the language model. In abstractive summarization, Li et al. (2019) have used BERTScore as reward in a deep reinforcement learning framework. In a similar vein, our work, too, aims to leverage pretrained language models for improving the NMT accuracy. However, to the best of our knowledge, ours is the first work to directly include a language model as a differentiable evaluation measure in the training objective. In this way, the NMT model is able to exploit the value of a pretrained language model while at the same time being fine-tuned over a task-specific evaluation metric.

3 BERTScore Optimization

Translation evaluation metrics, including BERTScore, typically require a predicted translation, $\langle \hat{y}_1, \ldots, \hat{y}_k \rangle$, and at least one reference translation, $\langle y_1, \ldots, y_l \rangle$, as inputs. At its turn, the predicted translation is typically obtained as a sequence of individual word (or token) predictions, using beam search or greedy decoding. We can express the predictions as:

$$\hat{y}_j = \arg\max_y p(y|x, \hat{y}_{j-1}, \theta) \quad j = 1, \ldots, k \quad (2)$$

where x represents the source sentence and θ the model's parameters. During model training, it is common practice to use teacher forcing (i.e., use words from the reference sentence as $\hat{y}_{j-1}$) for efficiency and faster convergence.

In brief, the computation of BERTScore works as follows: the scorer first converts the words in the predicted and reference sentences to corresponding static (i.e., non-contextual) word embeddings using the embedding matrix, $\mathbf{E}$, stored in the pretrained language model. For the predicted sequence, we note this lookup as:

$$\mathbf{e}_{\hat{y}_j} = \text{emb}_{LM}(\mathbf{E}, \hat{y}_j) \quad j = 1, \ldots, k \quad (3)$$

The sequences of static embeddings for the predicted and reference sentences are then used as

inputs into the language model to generate corresponding sequences of contextualized embeddings, $\langle \hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_k \rangle$ and $\langle \mathbf{y}_1, \ldots, \mathbf{y}_k \rangle$, respectively, over which the BERTScore is finally computed. For our work, we have chosen to optimize the F_{BERT} score as it balances precision and recall. For more details on the scoring function we refer the reader to (Zhang et al., 2020).

3.1 Soft predictions

However, it is not possible to directly use the F_{BERT} score as a training objective since the argmax function in (2) is discontinuous. Therefore, in this work we propose replacing the hard decision of the argmax with "soft predictions" that retain differentiability. Let us note concisely the probability in (2) as p_j^i, where i indexes a particular word in the V-sized vocabulary and j refers to the decoding step, and the entire probability vector at step j as $\mathbf{p}_j$. Let us also note as $\mathbf{e}^i$ the embedding of the i-th word in the embedding matrix of the pretrained language model, $\mathbf{E}$. We then compute an "expected embedding" as follows:

$$\bar{\mathbf{e}}_{\hat{y}_j} = \mathbb{E}[\mathbf{E}]_{\mathbf{p}_j} = \sum_{i=1}^{V} p_j^i \mathbf{e}^i \tag{4}$$

In other terms, probabilities $\mathbf{p}_j$ act as attention weights over the word embeddings in matrix $\mathbf{E}$, and the resulting expected embedding, $\bar{\mathbf{e}}_{\hat{y}_j}$, can be seen as a trade-off, or weighted average, between the embeddings of the words with highest probability. To be able to compute this expectation, the NMT model must share the same target vocabulary as the pretrained language model. Once the expected embeddings for the whole predicted sentence, $\langle \bar{\mathbf{e}}_{\hat{y}_1}, \ldots, \bar{\mathbf{e}}_{\hat{y}_k} \rangle$, are computed, they are input into the language model to obtain the corresponding sequence of predicted contextualized embeddings, and the F_{BERT} score is computed. The fine-tuning loss is simply set as $\mathcal{L} = -F_{BERT}$. During fine-tuning, only the parameters of the NMT model are optimized while those of the pretrained language model are kept unchanged.

3.2 Sparse soft predictions

A potential limitation of using the probability vectors to obtain the expected embeddings is that they are, a priori, dense, with several words in the vocabulary possibly receiving a probability significantly higher than zero. In this case, the expected embeddings risk losing a clear interpretation. While we could simply employ a softmax with temperature to sparsify the probability vectors, we propose exploring two more contemporary approaches:

- **Sparsemax** (Martins and Astudillo, 2016): Sparsemax generates a Euclidean projection of the logits computed by the decoder (noted as vector $\mathbf{s}_j$) onto the probability simplex, Δ^{V-1}:

$$\mathbf{p}_j^{SM} = \underset{\mathbf{p}_j \in \Delta^{V-1}}{\arg\min} ||\mathbf{p}_j - \mathbf{s}_j||^2 \tag{5}$$

  The larger the logits, the more likely it is that the resulting $\mathbf{p}_j^{SM}$ vector will have a large number of components equal to zero. The sparsemax operator is fully differentiable.

- **Gumbel-Softmax** (Jang et al., 2017; Maddison et al., 2017): The Gumbel-Softmax is a recent reparametrization technique that allows sampling *soft* categorical variables by transforming samples of a Gumbel distribution. The transformation includes a temperature parameter, τ, that allows making the resulting soft variables more or less sparse. By noting a sample from the Gumbel distribution as g^i, the Gumbel-Softmax can be expressed as:

$$p_j^{i\,GS} = \frac{\exp((\log p_j^i + g^i)/\tau)}{\sum_{v=1}^{V} \exp((\log p_j^v + g^v)/\tau)} \tag{6}$$

  where $p_j^{i\,GS}$, $i = 1, \ldots, V$, are the components of the probability vector used in (4). In the experiments, τ has been set to 0.1 to enforce sparsity. In addition to obtaining more "selective" predictions, the Gumbel-Softmax leverages sampling, allowing the fine-tuning to avail of a certain degree of *exploration*. The Gumbel-Softmax, too, is fully differentiable.

4 Experiments

4.1 Datasets

We have carried out multiple experiments over four, diverse language pairs, namely, German-English (de-en), Chinese-English (zh-en), English-Turkish (en-tr) and English-Spanish (en-es), using the datasets from the well-known IWSLT 2014 shared task[1], with 152K, 156K, 141K and 172K training sentences, respectively. Following Edunov et al. (2018), in the de-en dataset we have used $7,000$ samples of the training data for validation, and *tst2010*, *tst2011*, *tst2012*, *dev2010* and

[1] https://wit3.fbk.eu/2014-01

Model	de-en			zh-en			en-tr			en-es		
	BLEU	F_{BERT}	MS	BLEU	F_{BERT}	MS	BLEU	F_{BERT}	MS	BLEU	F_{BERT}	MS
Transformer NMT	33.61	77.56	52.86	18.28	68.04	34.81	17.68	76.55	18.3	37.80	79.31	45.76
+ BERTTune (DV)	33.58	77.90	53.4†	**18.53**†	**68.53**†	**35.57**†	17.81†	76.57	18.19	37.36	79.30	**45.92**†
+ BERTTune (SM)	33.39	77.88	53.27†	18.09	68.48†	35.18†	17.52	76.55	18.09	37.70	79.27	45.89†
+ BERTTune (GS)	**33.97**	**78.32**†	**53.58**†	18.39†	68.45†	35.33†	**18.26**†	**76.75**†	**18.33**	**37.96**†	**79.33**	45.84†

Table 1: Average BLEU, F_{BERT} and MoverScore (MS) results over the test sets. (†) refers to statistically significant differences with respect to the baseline computed with a bootstrap significance test with a p-value < 0.01 (Dror et al., 2018). The bootstrap test was carried out at sentence level for F_{BERT} and MS, and at corpus level for BLEU.

dev2012 as the test set. For the other language pairs, we have used the validation and test sets provided by the shared task. More details about the preprocessing are given in Appendix A.

4.2 Models and training

We have implemented the fine-tuning objective using the *fairseq* translation toolkit[2] (Ott et al., 2019). The pretrained language models for each language have been downloaded from Hugging Face (Wolf et al., 2020)[3]. As baseline, we have trained a full NMT transformer until convergence on the validation set. With this model, we have been able to reproduce or exceed the challenging baselines used in (Zhang et al., 2020; Xia et al., 2019; Miculicich et al., 2018; Wu et al., 2020). The fine-tuning with the F_{BERT} loss has been carried out over the trained baseline model, again until convergence on the validation set. For efficient training, we have used teacher forcing in all our models. During inference, we have used beam search with beam size 5 and length penalty 1. As performance measures, we report the BLEU, F_{BERT} and MoverScore (MS) (Zhao et al., 2019) results over the test sets averaged over three independent runs. Including BLEU and MS in the evaluation allows us to probe the models on metrics different from that used for training. Similarly to F_{BERT}, MS, too, is a contextual embedding distance-based metric, but it leverages soft alignments (many-to-one) rather than hard alignments between words in the candidate and reference sentences. To make the evaluation more probing, for MS we have used different pretrained language models from those used with F_{BERT}. For more details on the models and hyperparameter selection, please refer to Appendix A.

[2]https://github.com/ijauregiCMCRC/fairseq-bert-loss
[3]https://huggingface.co/models

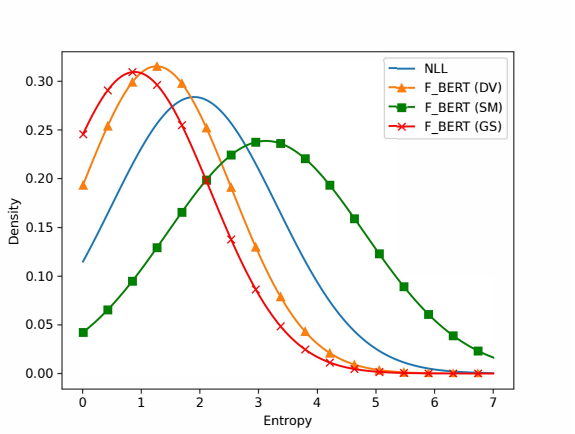


Figure 1: Entropy of the probability vectors generated by the different approaches over the de-en test set.

4.3 Results

Table 1 shows the main results over the respective test sets. As expected, fine-tuning the baseline with the proposed approach has generally helped improve the F_{BERT} scores. However, Table 1 also shows that it has often led to improvements in BLEU score. In the majority of cases, the best results have been obtained with the Gumbel-Softmax (GS), with more marked improvements for de-en and en-tr ($+0.36$ pp BLEU and $+0.76$ pp F_{BERT} and $+0.72$ pp MS for de-en, and $+0.58$ pp BLEU, $+0.20$ pp F_{BERT} and $+0.03$ pp MS for en-tr). Conversely, the dense vectors (DV) and sparsemax (SM) have not been as effective, with the exception of the dense vectors with the zh-en dataset ($+0.25$ pp BLEU, $+0.49$ pp F_{BERT} and $+0.54$ pp MS). This suggests that the Gumbel-Softmax sampling may have played a useful role in exploring alternative word candidates. In fairness, none of the proposed approaches has obtained significant improvements with the en-es dataset. This might be due to the fact that the baseline is much stronger to start with, and thus more difficult to improve upon. In general, both the embedding-based metrics (i.e., F_{BERT} and MS) have ranked the approaches in the same order, with the exception of the en-es dataset.

To provide further insights, similarly to Baziotis et al. (2020), in Figure 1 we plot the distribution of the entropy of the probability vectors generated by the different approaches during inference over the de-en test set. Lower values of entropy correspond to sparser predictions. The plot shows that the models fine-tuned with the dense vectors and the Gumbel-Softmax have made test-time predictions that have been sparser on average than those of the baseline, with the Gumbel-Softmax being the sparsest, as expected. Conversely, and somehow unexpectedly, the model fine-tuned with the sparsemax has made predictions denser than the baseline's. We argue that this may be due to the scale of the logits that might have countered the aimed sparsification of the sparsemax operator. In all cases, the sparsity of the predictions seems to have positively correlated with the improvements in accuracy. For a qualitative analysis, Appendix B presents and discusses various comparative examples for different language pairs.

Finally, Figure 2 shows the effect of the proposed objective over the measured metrics on the de-en validation set at different fine-tuning steps. The plots show that the model rapidly improves the performance in F_{BERT} and MS scores during the first epoch (steps $1 - 967$), peaking in the second epoch ($\approx$ step $1,200$). After that, the performance of the model starts dropping, getting back to the baseline levels in epoch 4. This suggests that training can be limited to a few epochs only, to prevent overfitting. On the other hand, the plots also show a trade-off between the metrics, as the model's improvements in F_{BERT} and MS come at cost of a decrease in BLEU. However, this phenomenon has not been visible on the test set, where all the fine-tuned models have outperformed the baseline also in BLEU score. This suggests that for this dataset the distributions of the training and test sets may be more alike.

5 Conclusion

In this work, we have proposed fine-tuning NMT models with BERTScore, a recently proposed word embedding-based evaluation metric aimed to overcome the typical limitations of n-gram matching. To be able to use BERTScore as an objective function while keeping the model end-to-end differentiable, we have proposed generating *soft* predictions with differentiable operators such as the sparsemax and the Gumbel-Softmax. The experimental results over four language pairs have showed that the proposed approach – nicknamed *BERTTune* – has been able to achieve statistically significant improvements in BLEU, F_{BERT} and MS scores over a strong baseline. As future work, we intend to explore the impact of key factors such as the dataset size, the sparsity degree of the predictions and the choice of different pretrained language models, and we also plan to evaluate the use of beam search/sequential sampling during training to leverage further exploration of candidate translations.

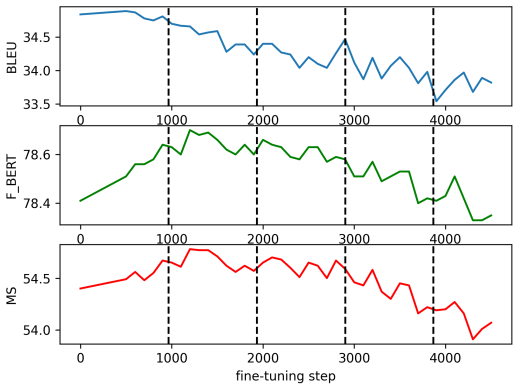


Figure 2: BLEU, F_{BERT} and MS scores of the BERTTune (GS) model over the de-en validation set at different fine-tuning steps. Step 0 is the score of the baseline model, and the vertical dashed lines delimit the epochs.

References

Christos Baziotis, Barry Haddow, and Alexandra Birch. 2020. Language model prior for low-resource neural machine translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*.

Katsuki Chousa, Katsuhito Sudoh, and Satoshi Nakamura. 2018. Training neural machine translation using word embedding-based loss. *arXiv preprint arXiv:1807.11219*.

Stéphane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of bert for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statis-

tical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1383–1392.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2018. Token-level and sequence-level loss smoothing for RNN language models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2094–2103.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations*.

Inigo Jauregi Unanue, Ehsan Zare Borzeshi, Nazanin Esmaili, and Massimo Piccardi. 2019. ReWE: Regressing word embeddings for regularization of neural machine translation systems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 430–436.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Sachin Kumar and Yulia Tsvetkov. 2019. Von mises-fisher loss for training sequence to sequence models with continuous outputs. In *Proceedings of the International Conference on Learning Representations*.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018. Bag-of-words as target for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 332–338.

Christopher Maddison, Andriy Mnih, and Yee Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the International Conference on Learning Representations*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, pages 5998–6008.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1296–1306.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Lijun Wu, Shufang Xie, Yingce Xia, Yang Fan, Jian-Huang Lai, Tao Qin, and Tie-Yan Liu. 2020. Sequence generation with mixed representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 10388–10398.

Yingce Xia, Tianyu He, Xu Tan, Fei Tian, Di He, and Tao Qin. 2019. Tied transformers: Neural machine translation with shared encoder and decoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5466–5473.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9378–9385.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of the International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 563–578.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. In *Proceedings of the International Conference on Learning Representations*.

Appendix A: Preprocessing and hyperparameters

This appendix provides detailed information about the preprocessing of the datasets and the hyperparameter selection to facilitate the reproducibility of the experiments. All the code will be released after the anonymity period.

As part of the preprocessing of the datasets, all sentences have been tokenized and lowercased. The source languages have been tokenized with the Moses tokenizer[4], except for Chinese that has been tokenized using Jieba[5]. The target languages have instead been tokenized with the tokenizer learned by the pretrained language model. As language models for BERTScore, we have used `bert-base-uncased` (en), `dbmdz/bert-base-turkish-uncased` (tr) and `dccuchile/bert-base-spanish-wwm-uncased` (es) from Hugging Face. As language model for the MoverScore, we have used the suggested language model for English[6], `dbmdz/distilbert-base-turkish-cased` for Turkish and `mrm8488/distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es` for Spanish, the last two from Huggingface. The few sentences longer than 175 tokens have been removed from all datasets as in the original fairseq preprocessing script. Additionally, further tokenization at subword level has been performed over the source languages using byte-pair encoding (BPE) (Sennrich et al., 2016) with $32,000$ merge operations. An important step in the preprocessing has been to force the decoder and the language model to share the same vocabulary. Therefore, we have assigned the decoder with the vocabulary from the selected pretrained language model, ensuring that both used identical `bos`, `eos`, `pad` and `unk` tokens.

For training a strong transformer baseline, we have followed the recommendations in fairseq[7]. The architecture is the predefined `transformer_iwslt_de_en` architecture (79M parameters) with word embedding and hidden vector dimension size of 512, and 6 transformer layers. We have set the training batch size to $4,096$ tokens, the dropout rate to 0.3 and the `clip_norm` gradient clipping parameter to 0.0. The objective function is the label-smoothed negative log-likelihood, with the smoothing factor set to 0.1. We have used the Adam optimizer (Kingma and Ba, 2015) with a $5e-4$ learning rate and beta values $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We have set the warm-up steps to $4,000$ with an initial learning rate of $1e-7$. During training, we have reduced the learning rate with an inverse square-root scheduler and the weight decay set to 0.001. We have trained the model until convergence of the BLEU score on the validation set, with checkpoints at each epoch and patience set to 3, or until the learning rate dropped below $1e-9$.

For fine-tuning with F_{BERT}, we have initialized the transformer models with the trained weights of the baseline. We have kept all hyperparameters identical, except for the learning rate which has been reduced by an order of magnitude to $5e-5$, following common fine-tuning strategies. The models have been fine-tuned until convergence over the validation set, with patience set to 3. Since the changes have only involved the training objective, the number of trainable parameters has remained exactly the same (79M). At test time, we have used beam search decoding with beam size 5 and length penalty 1.

For all the experiments we have used an NVIDIA Quadro P5000 GPU card with 16 GB of memory.

Appendix B: Translation examples

This appendix shows a few translation examples from the de-en and zh-en language pairs to provide further insights into the behavior of the different models.

The example in Table 2 shows that only the BERTTune model with the Gumbel-Softmax has been able to translate phrases such as *at the moment* and *it was as if / it was like*. This model seems to have been able to capture the exact meaning of the source German sentence, even though it has translated it with a slightly different wording (note that the Gumbel-Softmax fine-tuning explores a larger variety of predictions). The other BERTTune models, too, have translated this sentence better than the baseline.

In the example in Table 3, the baseline has not been able to correctly pick the name of the artist (*bono*, lowercased from *Bono*), choosing instead word *bonobos* (primates). All the BERTTune mod-

[4] https://github.com/moses-smt/mosesdecoder
[5] https://github.com/fxsjy/jieba
[6] https://github.com/AIPHES/emnlp19-moverscore/releases/download/0.6/MNLI_BERT.zip
[7] https://fairseq.readthedocs.io/en/latest/index.html

els have instead made the correct prediction. In this example, it is possible that the BERTTune models have benefited from the fine-tuning with a pretrained language model: word *bono* might not have been present in the limited translation training data, but might have been encountered in the large unsupervised corpora used to train the language model. Another possibility is that they have simply used the copy mechanism more effectively.

In the example in Table 4, all the BERTTune models have correctly translated the phrase *part of the national statistics*, while the baseline has incorrectly translated it as *part of the world record*. In turn, the BERTTune models have translated the phrase *in a decade or two* as *in 10 or 20 years* which is a correct paraphrase, whereas the baseline has used the exact phrase as the reference. We also note that although both the baseline and BERTune translations have scored a BLEU score of 0.0 in this case, the F_{BERT} score has been able to differentiate between them, assigning a score of 72.36 to the BERTTune translation and 72.10 to the baseline. This also shows that small gains in F_{BERT} score can correspond to significant improvements in translation quality.

Finally, in the example in Table 5 only the BERTTune models with dense vectors and Gumbel-Softmax have been able to translate the beginning of the sentence (*i was the guy beaten up*) with acceptable paraphrases (i.e. *and i was the kind of person who had been beaten up / i was that guy who had been beaten*). Conversely, the baseline has translated the ending part of the sentence (*until one teacher saved my life*) with a phrase of antithetical meaning (*until a teacher turned me into this kind of life*).

Src:	in dem moment war es , als ob ein filmregisseur einen bühnenwechsel verlangt hätte .
Ref:	at that moment , it was as if a film director called for a set change .
Transformer NMT:	the moment a film director would have asked a stager .
BERTTune (Dense vectors)	and at that moment , a film director would have wanted a stage change .
BERTTune (Sparsemax)	the moment a film director wanted a stage change .
BERTTune (Gumbel-Softmax)	at the moment , it was like a film director would have wanted a stage change .

Table 2: De-en translation example.

Src:	und interessanterweise ist bono auch ein ted prize gewinner .
Ref:	and interestingly enough , bono is also a ted prize winner .
Transformer NMT:	and interestingly , bonobos are also a ted prize winner .
BERTTune (Dense vectors)	and interestingly , bono is also a ted prize winner .
BERTTune (Sparsemax)	and interestingly , bono is also a ted prize winner .
BERTTune (Gumbel-Softmax)	and interestingly , bono is also a ted prize winner .

Table 3: Another de-en translation example.

Src:	我想在十年或二十年内，这将会成为国家统计数据的一部分。
Ref:	this is going to be , i think , within the next decade or two , part of national statistics .
Transformer NMT:	i think it ' s going to be part of the world record in a decade or two .
BERTTune (Dense vectors)	and i think that in 10 or 20 years , this will be part of the national statistics .
BERTTune (Sparsemax)	i think that in 10 or 20 years , this will be part of the national statistics .
BERTTune (Gumbel-Softmax)	i think that in 10 or 20 years , this will be part of the national statistics .

Table 4: Zh-en translation example.

Src:	我是那种每周在男生宿舍被打到出血的那种人直到一个老师把我从这种生活中解救出来。
Ref:	i was the guy beaten up bloody every week in the boys ' room , until one teacher saved my life .
Transformer NMT:	i was the one who was in the dorm room every week , and it wasn ' t until a teacher turned me into this kind of life .
BERTTune (Dense vectors)	and i was the kind of person who had been beaten up in the dorma every week until a teacher turned me out of this life .
BERTTune (Sparsemax)	i ' m the kind of person who fell into his dorm room till a teacher turned me through this kind of life .
BERTTune (Gumbel-Softmax)	i was that guy who had been beaten in his dorm room every week, until a teacher took me out of that life .

Table 5: Another zh-en translation example.

Entity Enhancement for Implicit Discourse Relation Classification in the Biomedical Domain

Wei Shi[§,†] and **Vera Demberg**[†,‡]
[§] Alibaba Group, Hangzhou, China
[†] Dept. of Language Science and Technology
[‡] Dept. of Mathematics and Computer Science, Saarland University
Saarland Informatics Campus, Saarbrücken, Germany
{w.shi, vera}@coli.uni-saarland.de

Abstract

Implicit discourse relation classification is a challenging task, in particular when the text domain is different from the standard Penn Discourse Treebank (PDTB; Prasad et al., 2008) training corpus domain (Wall Street Journal in 1990s). We here tackle the task of implicit discourse relation classification on the biomedical domain, for which the Biomedical Discourse Relation Bank (BioDRB; Prasad et al., 2011) is available. We show that entity information can be used to improve discourse relational argument representation. In a first step, we show that explicitly marked instances that are content-wise similar to the target relations can be used to achieve good performance in the cross-domain setting using a simple unsupervised voting pipeline. As a further step, we show that with the linked entity information from the first step, a transformer which is augmented with entity-related information (K-BERT; Liu et al., 2020) sets the new state of the art performance on the dataset, outperforming the large pre-trained BioBERT (Lee et al., 2020) model by 2% points.

1 Introduction

Discourse relation classification (DRC) involves automatically inferring the logical link between different text segments (such as causal, contrastive, temporal etc.). It has been shown to be a valuable preprocessing step to many downstream natural language processing tasks such as machine translation (Guzmán et al., 2014; Meyer et al., 2015), text summarization (Gerani et al., 2014) and question-answering (Jansen et al., 2014). A main obstacle to a wider usage of automatic DR classifiers however lies in getting the classifiers to work reliably on domains other than the WSJ, that discourse relation parsers are usually trained on PDTB (Prasad et al., 2008) and RST (Carlson et al., 2003).

Moving to a different domain is particularly challenging in DRC because the overall distribution of relations typically differs between domains, and because many of the content words that classifiers may rely on are very different between domains. We here focus on the most challenging subtask of *implicit discourse relation classification*, which involves classifying those relations that are not linked by any explicit connectives like "because" or "but". In order to correctly recognize implicit relations, the classifier needs to recognize subtle surface cues (which may differ between domains) and learn about typical content-related relations. For instance, from the example "it's hot outside, therefore I'd like to eat an icecream", the words "hot outside" and "icecream" are relevant cues for the relation. An overview of typical cues for determining a coherence relation is provided in Das and Taboada (2018).

The key to improving automatic DRC on a new domain hence consists of better encoding of the discourse relational arguments. As we will show below (in line with earlier findings by Shi and Demberg, 2019b), it makes a big difference to have at least a small amount of in-domain discourse annotated data.

We here explore DRC on the biomedical domain, which seems particularly suitable because a discourse-annotated corpus is available (BioDRB; Prasad et al., 2011), which we can use for evaluation, as well as a setting with a small amount of in-domain training data. Furthermore, the biomedical domain does have large raw text corpora available. An example instance from BioDRB (Prasad et al., 2011) is shown below:

1. *[These abnormalities in active RA are thought to be induced mainly after chronic exposure to high concentrations of IL-6.]*$_{Arg1}$ *(**Implicit=thus**) [The limited efficacy of IL-10*

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 925–931
August 1–6, 2021. ©2021 Association for Computational Linguistics

treatment of RA patients may be explained in part by the unresponsiveness to IL-10 of inflammatory cells, including T cells .]$_{Arg2}$

—Implicit, Contingency.Cause

Scientific texts such as those from the biomedical domain are well known to express much of the content in nominal phrases, and less in verb phrases (Halliday, 2006). Concretely, for the above example, understanding the relation between the RA (Rheumatoid Arthritis) and inflammatory cells (including T cells) is important to correctly understanding the relation. The high importance of entities in these texts is a crucial insight on which we base our approach.

In this paper, we first propose an unsupervised method using information retrieval and knowledge graph techniques for identifying text passages that are similar content-wise to the coherence relation we want to label. The underlying assumption here is that if two instances share the same entities in both the relational arguments, it is possible that they have the same or a similar discourse relation. This part of the method is applicable to any domain for which large amounts of in-domain text are available, but no in-domain discourse relation annotations. We find that this method helps to improve results substantially compared to a Bi-LSTM baseline model, but doesn't reach state of the art performance (which is set by transformer models).

We therefore proceed to enrich a transformer model with the knowledge extracted from the unlabelled texts, using the K-BERT model (Liu et al., 2020). The model is fine-tuned on the discourse-annotated in-domain BioDRB data. We show that this setting sets the new state of the art on discourse relation classification on the biomedical domain, achieving an accuracy of 69.57%.

2 Related Work

Early approaches on BioDRB use probabilistic classifiers such as Naïve Bayes, Maximum Entropy, etc. to predict the relation (Xu et al., 2012). Bai and Zhao (2018) combine representations from different types of embeddings including contextualized word vectors from ELMo (Peters et al., 2018) and achieve 55.9% accuracy on BioDRB for in-domain training, and 29.52% in the cross-domain setting (reported in Shi and Demberg (2019b)).

Shi and Demberg (2019b) also explore the performance of BERT (Devlin et al., 2019) models on the DRC task on BioDRB using cross-domain (fine-tuning on PDTB, testing on BioDRB) as well as in-domain (fine-tuning on BioDRB and testing on BioDRB) settings. They find a very good performance of the BERT model, which they attribute to its "next sentence prediction" task in pre-training. Comparing the original BERT model to BioBERT (Lee et al., 2020), which was trained on biomedical text, they however find that BioBERT has only a limited ability for learning domain specific representations: Cross-domain performance is no better than for the BERT model, and in-domain performance improvements are moderate at only 1.5% points. Given that the entities play an important role in inferring implicit discourse relation in scientific texts, putting an emphasis on entities seems vital for achieving further improvements.

In contrast with previous studies that (largely unsuccessfully) attempted to train on explicit discourse relations for learning to classify implicit classifiers in supervised ways, such as Marcu and Echihabi (2002); Sporleder and Lascarides (2008); Biran and McKeown (2013); Qin et al. (2017); Shi et al. (2017) etc., we here propose an unsupervised voting pipeline and achieve good performance even comparing with supervised models like BERT and BioBERT. We believe that the key difference lies in the fact that previous methods tried to learn *surface cues* from explicit relations and tried to use them for implicits (which does not work, because these features differ between explicits and implicits, see e.g., Sporleder and Lascarides (2008); Asr and Demberg (2012)), while our method focuses on the content of the discourse relational arguments.

3 Unsupervised Method with Information Retrieval System

The successful usage of a memory network in Shi and Demberg (2019a) showed that instances that share the same relation have close representations. We believe that for sparse data like BioDRB, which has only around 2,000 labeled implicit instances in total, it is essential to use similar explicit instances to help find the latent patterns they share. In this section, we introduce an unsupervised method for implicit DRC, which is inspired by a recent information retrieval method.

The core idea is as follows: we use information retrieval methods to identify explicitly marked coherence relations from the corpus which are content-wise similar to the relation we want to la-

bel. We then automatically label these explicitly marked instances (relying on the high DRC accuracy of ca. 96% for explicit relations) and assign the majority label from the explicit instances to the implicit instance from our test set.

3.1 Retrieval of similar instances from a large corpus

Figure 1 illustrates the overall pipeline of the proposed method. First, each instance from BioDRB (Prasad et al., 2011) is seen as a query and fed into the PubMed[1] and PMC[2] databases.

PubMed and **PMC** are free full-text archives of biomedical and life sciences journal literature at NIH National Library of Medicine. The database we use here is a corpus created from a subset of the whole PubMed and PMC collections, consisting of 7,079 documents in total (1,376 for pubMed and 5,703 for PMC).

With the query and candidate documents, we employ TF-IDF to extract the top 10 relevant documents. The candidate documents are then fed into a discourse parser; we here use the PDTB-style end-to-end parser by Lin et al. (2014). The outputs of the parser contain the two arguments, the explicit discourse connective and a discourse relation label.

The Quasi Knowledge Graphs System, proposed by Lu et al. (2019), is designed to answer complex questions. It is a novel method that computes answers by dynamically building up a knowledge graph that fits the query. It consists of several steps including the extraction of subject-predicate-object (SPO) triples, knowledge graph construction, and a graph algorithm. We here only use the first step from this pipeline, extracting SPO triples, and actually only use the subject and object, not the predicate, to match with the noun phrases in the query. For example, from the relation instance in Example 1 above, the system would extract SPO triples *(NETosis, enhanced in, RA)* and *(autoantibodies, known risk factors for, RA)*, from which we further employ only *NETosis, RA; autoantibodies, RA*.

After extracting the SPO triples from all the explicit discourse instances, we employ two types of matching strategies to connect them with the query:

[1]PubMed [Internet]. Bethesda (MD): National Library of Medicine (US). [1946]. Available from: https://www.ncbi.nlm.nih.gov/pubmed/

[2]PubMed Central (PMC) [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; 2000. Available from: https://www.ncbi.nlm.nih.gov/pmc/

Methods	Cross-domain
Majority class	20.66
Bai and Zhao (2018)	29.52
Bi-LSTM + Word2Vec	32.97
BERT	44.79
BioBERT	44.33
Hard-matching	35.29
Soft-matching	41.95

Table 1: Performances on BioDRB across domains. Across domains means that the model is trained on PDTB and tested on BioDRB. Majority class here is the majority relation of explicits.

(i) **Hard matching**, which means that if the subject or object appear in the query, we count it as a vote. (ii) **Soft matching**. We find that with the hard matching, lots of positive samples have been filtered out and very few explicit instances are identified. Therefore, we use the cosine similarity between the subject or object and the noun phrases in the query, to detect similar entities. Cosine similarities are estimated based on the BioBERT encoding of the entities. We define a threshold for deciding when an explicit instance is similar enough to be counted as a valid vote or not. It is seen in the training phase as a hyper-parameter to be fine-tuned on the validation set. This method for detecting similar explicit instances is also used in our second approach described in Section 4.

With the steps described above, eventually each query has been connected to a number of similar explicit instances and the prediction for the query is the majority vote from all of them with their explicit discourse sense labels.

3.2 Experiments and results

On average 813.99 explicit instances are extracted for each query. With the hard matching, 7.91 similar entities are matched with the Subject or the Object in the query. For the soft matching, we randomly choose 10% of the total instances acting as validation set in order to help set the threshold for the cosine similarity score.

The experimental results are shown in Table 1. We compare the results with related work by Bai and Zhao (2018) as well as several models reported in Shi and Demberg (2019b).

Our proposed unsupervised method achieves an accuracy of 35.29% with hard-matching and 41.95% with soft-matching. These results outper-

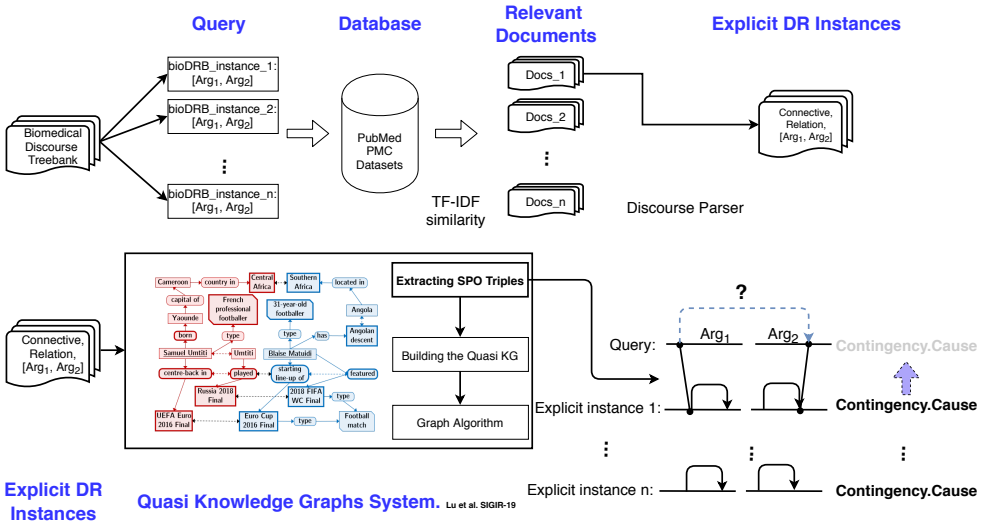


Figure 1: The Pipeline of the Proposed Method.

form other non-transformer approaches by a large margin. Comparing the hard and soft matching variants, our results show that identifying instances with similar entities leads to a larger set of relevant documents, which then help to increase robustness in the majority vote.

The table also shows that the approach almost reaches the performance of recent very strong transformer models: the BERT model achieves a performance of 44.79% accuracy in the cross-domain setting (Shi and Demberg, 2019b).

The approach proposed here could be further refined by using better argument representations than simple matching of subject and object entities, and by learning the classification decisions instead of using simple majority voting, and by moving to transformer architectures. Our second approach addresses these points by employing a transformer architecture which can take the SPO triple information into account for more richly encoding the relational arguments.

4 DRC with an entity-augmented transformer

Integrating external domain-specific knowledge into the model is beneficial for this task has been found by Kishimoto et al. (2018), who integrated the ConceptNet relations as additional knowledge into the LSTM network and achieved better performance on the PDTB.

We here aim to explore whether model performance can be further improved by exploiting richer entity representations in specialized texts

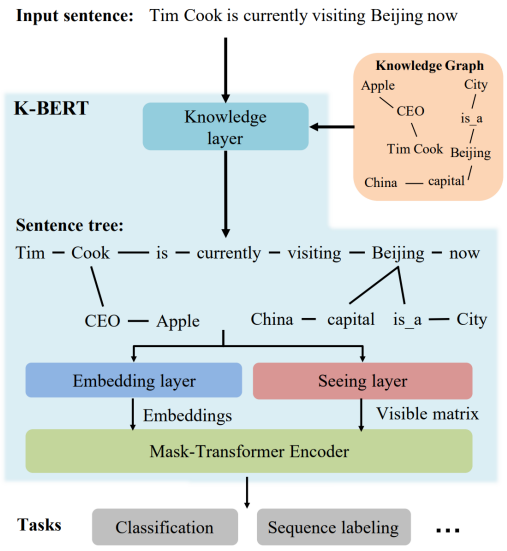


Figure 2: The structure of K-BERT. It is equipped with an editable knowledge graph which can be adapted to its application domain. Picture taken from Liu et al. (2020).

like the biomedical domain. The pipeline with soft-matching proposed in the above section provides us with SPO triples from related documents for each implicit relation instance in the test set. We here employ the recently proposed **Knowledge-enabled Language Representation model** (Liu et al., 2020, K-BERT) to integrate the external entity knowledge into the pre-trained language model for better argument representations.

4.1 K-BERT

Due to the domain gap between the pre-training and fine-tuning, unsupervised language models (such

as BERT etc.) do not perform well on knowledge-driven tasks (Liu et al., 2020). Integrating domain specific knowledge into pre-trained model can alleviate this problem. However, the process of knowledge acquisition can be inefficient and expensive.

In order to tackle the heterogeneous embedding space and knowledge noise problems, Liu et al. (2020) proposed a Knowledge-enabled Bidirectional Encoder Representation from Transformers (K-BERT), as illustrated in Figure 2. With the knowledge layer and the external knowledge graph, the input sentence has been expanded into a sentence tree, which is then fed into into the embedding layer and the "seeing" layer. The seeing layer controls when the model has access to the original sentence and when it has access to the additional information.

However, knowledge graphs are not available for all domains. We therefore here replace information from the knowledge graph with the SPO triples extracted from related raw texts. Compared to a general knowledge graph, our extracted SPO triples have attached more importance on the discourse relations since that they are extracted from the explicit instances, and are specifically selected to be on-topic. For each input sentence, we attach the top 2 (default number from the K-BERT) similar SPO triples to the entities and convert it into a sentence tree. We train K-BERT on the BioDRB as a classification task. The input sequence of the Example 1 is shown below, where the words in italics are the linked entities.

2. These abnormalities in active *NETosis enhanced in autoantibodies known risk factors for* RA *result in Neutrophil Chemotaxis* are thought to be induced mainly after chronic exposure to high concentrations of IL-6. The limited efficacy of IL-10 treatment of RA patients *reduced complement activation* may be explained in part by the unresponsiveness to IL-10 of inflammatory cells, including T cells *isolated from CTCL patient*.

The whole sentence tree has been flattened into a sequence with the position index. The visible matrix is generated to keep the interactions of each of the tokens within the original sentence and also inside the knowledge graph triples. The visible matrix controls the self-attention layers in the transformer not to look into tokens other than the corresponding entities.

Methods	In-domain
Bai and Zhao (2018)	55.90
Bi-LSTM + Word2Vec	46.49
BERT	63.02
BioBERT	67.58
proposed model using K-BERT	**69.57***

Table 2: Performances on BioDRB within domain. Within domain here means 5-folds cross validation (see also Shi and Demberg (2017)) on BioDRB. * denotes significant improvement over BioBERT with $p<0.05$.

4.2 Experiments and Results

The experimental results are illustrated in Table 2. We compare the results with the previous state of the art on the BioDRB dataset (Shi and Demberg, 2019b). K-BERT, which is initialized with the original BERT parameters, achieves 69.57% accuracy and outperforms BERT without entity augmentation by 6.5% points, and the the gigantic in-domain continuously pre-trained BioBERT by around 2%. In addition, we tried to remove the relevant entities. The model then performed similar to the basic BERT, which is consistent with the results reported in Liu et al. (2020). These results confirm that adding related entities improves argument encoding and help improve the DRC task.

5 Conclusion

In this paper, we address the task of implicit discourse relation classification on BioDRB in the biomedical domain. Due to the importance of entities in scientific text, we decided to address this problem by identifying explicitly marked relations containing the same instances, and using a simple majority voting system. While this setting showed good performance in the unsupervised setting, much better results are achieved when at least a small amount of labelled data is available. We show that when a transformer model is augmented with entity information from the domain, the previous state of the art on the task is exceeded by 2% points.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable and constructive feedback. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.

References

Fatemeh Torabi Asr and Vera Demberg. 2012. Implicitness of discourse relations. In *Proceedings of COLING 2012*, pages 2669–2684.

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 571–583.

Or Biran and Kathleen McKeown. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 69–73.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer.

Debopam Das and Maite Taboada. 2018. Rst signalling corpus: A corpus of signals of coherence relations. *Language Resources and Evaluation*, 52(1):149–184.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1602–1613. Association for Computational Linguistics.

Francisco Guzmán, Shafiq R. Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using discourse structure improves machine translation evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 687–698. Association for Computational Linguistics.

Michael Alexander Kirkwood Halliday. 2006. *Language of science*, volume 5. Bloomsbury Publishing.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 977–986.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2018. A knowledge-augmented neural network model for implicit discourse relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 584–595.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI*, pages 2901–2908.

Xiaolu Lu, Soumajit Pramanik, Rishiraj Saha Roy, Abdalghani Abujabal, Yafang Wang, and Gerhard Weikum. 2019. Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 105–114.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 368–375.

Thomas Meyer, Najeh Hajlaoui, and Andrei Popescu-Belis. 2015. Disambiguating discourse connectives for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(7):1184–1197.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC08)*.

Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. 2011. The biomedical discourse relation bank. *BMC bioinformatics*, 12(1):188.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 1006–1017.

Wei Shi and Vera Demberg. 2017. On the need of cross validation for discourse relation classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 150–156.

Wei Shi and Vera Demberg. 2019a. Learning to explicitate connectives with Seq2Seq network for implicit discourse relation classification. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 188–199, Gothenburg, Sweden. Association for Computational Linguistics.

Wei Shi and Vera Demberg. 2019b. Next sentence prediction helps implicit discourse relation classification within and across domains. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5794–5800.

Wei Shi, Frances Yung, Raphael Rubino, and Vera Demberg. 2017. Using explicit discourse connectives in translation for implicit discourse relation classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 484–495.

Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369.

Yu Xu, Man Lan, Yue Lu, Zheng Yu Niu, and Chew Lim Tan. 2012. Connective prediction using machine learning for implicit discourse relation classification. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Unsupervised Pronoun Resolution via Masked Noun-Phrase Prediction

Ming Shen* **Pratyay Banerjee*** **Chitta Baral**
Arizona State University
mshen16, pbanerj6, chitta@asu.edu

Abstract

In this work, we propose Masked Noun-Phrase Prediction (MNPP), a pre-training strategy to tackle pronoun resolution in a fully unsupervised setting. Firstly, We evaluate our pre-trained model on various pronoun resolution datasets without any finetuning. Our method outperforms all previous unsupervised methods on all datasets by large margins. Secondly, we proceed to a few-shot setting where we finetune our pre-trained model on WinoGrande-S and XS separately. Our method outperforms RoBERTa-large baseline with large margins, meanwhile, achieving a higher AUC score after further finetuning on the remaining three official splits of WinoGrande.

1 Introduction

Co-reference resolution is an important NLP task that aims to find all expressions that refer to the same entity in a text. The resolution of an ambiguous pronoun, known as pronoun resolution, is a longstanding challenge for the NLU community and an essential step for various high-level NLP tasks such as natural language inference (Bowman et al., 2015; Williams et al., 2018), question answering (Rajpurkar et al., 2016), and relation extraction (Zhang et al., 2017).

The most successful approach to pronoun resolution is first fine-tuning a large pre-trained language model such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) on a human-labeled pronoun resolution dataset such as Definite Pronoun Resolution Dataset (DPR) (Rahman and Ng, 2012) or WinoGrande (WG) (Sakaguchi et al., 2020), and then either directly transferring to a smaller dataset such as Winograd Schema Challenge (WSC) (Levesque et al., 2012) or Pronoun Disambiguation Problems (PDP) (Morgenstern et al., 2016) or further finetuning on a downstream dataset such as SuperGLUE-WSC (Wang et al., 2019a). However, all the pipelines above can not avoid the phase of pre-training on a large human-labeled pronoun resolution dataset. Crowd-sourced "unbiased" labels that do not introduce annotation-artifacts (Gururangan et al., 2018) are shown to be costly and challenging to collect, requiring a well-designed annotation interface and dedicated annotators. To this end, we propose the unsupervised Masked Noun-Phrase Prediction task to pre-train a language model without any pronoun resolution training signal and directly transfer the pre-trained model to downstream datasets such as WSC.[1] Two examples of WSC are listed in Table 1. Our work improves on all previous unsupervised methods by large margins and even outperforms several strong supervised methods on all datasets we study.

WSC Sentences	Candidate Choices
The trophy doesn't fit in the suitcase because **it** is too small.	**A.** the trophy **B.** the suitcase
The trophy doesn't fit in the suitcase because **it** is too big.	**A.** the trophy **B.** the suitcase

Table 1: Above are two WSC examples. A system is required to resolve the bold pronoun **"it"** to "the suitcase" in the first sentence and to "the trophy" in the second sentence.

We then proceed to the few-shot setting where we finetune our best zero-shot model on WinoGrande-S and XS respectively. MNPP gives a large margin of improvements over strong baselines including CSS (Klein and Nabi, 2020), RoBERTa-large (Sakaguchi et al., 2020), and UnifiedQA-BART-large (Khashabi et al., 2020). We further finetune on the remaining three data splits and achieve a higher AUC score on all five splits of WinoGrande over RoBERTa-large baseline.

*Equal Contribution

[1] We refer to unsupervised or zero-shot transfer as without training on any pronoun resolution dataset.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 932–941
August 1–6, 2021. ©2021 Association for Computational Linguistics

In summary, our main contributions in this work are threefold.

- **First**, we propose the MNPP pre-training task and study how different synthetic dataset properties affect zero-shot performances.
- **Second**, we show MNPP outperforms all previous fully unsupervised methods and even several strong supervised baselines on all pronoun resolution datasets we study.
- **Finally**, we show that under few-shot settings, MNPP pre-training gives a significant performance boost on WinoGrande-S and XS and furthermore achieves a higher AUC score over all five splits of WinoGrande.

2 Related Works

In this work, we mainly compare with unsupervised methods.[2] On WSC, Zhang and Song (2018) propose the first unsupervised model where they modify Skip-Gram (Mikolov et al., 2013) objective to predict semantic dependencies then use this additional information during testing. Wang et al. (2019b) propose Unsupervised Deep Structured Semantic Models (UDSSM), which utilizes BiLSTM (Hochreiter and Schmidhuber, 1997) to compute contextual word embedding and uses models ensemble. Klein and Nabi (2019) directly explore the inner attention layers of BERT. Ye et al. (2019) adapt a masking and predicting strategy, called align, mask, and select (AMS), where entities that are connected with ConceptNet (Speer and Havasi, 2012) are masked and the model is required to select from a given list of candidate entities. An ensemble of large pre-trained models is first utilized by Trinh and Le (2018). GPT-2 is directly evaluated on WSC in Radford et al. (2019). Prakash et al. (2019) extend a language model with a knowledge hunting strategy. Kocijan et al. (2019b) and Kocijan et al. (2019a) are the most similar works to us and we will discuss the details in Section 3.1. Most recently, Klein and Nabi (2020) study a contrastive self-supervised learning approach (CSS) for WSC and DPR and also establish the first unsupervised baseline for KnowRef (Emami et al., 2019). On WinoGrande, knowledge hunting (Prakash et al., 2019) and language models ensemble (Sakaguchi et al., 2020) have been studied.

[2]Please refer to supplemental materials for more details on supervised methods.

3 Masked Noun-Phrase Prediction

We treat MNPP as a binary classification task. Given the sentence: *"She put the cup on the chair, but he knocked over the chair, and the cup fell."*, the underlined *"the chair"* will be masked and a pair of replacement phrases for this masked position is given as {*"the cup"*, *"the chair"*}. One of the candidates is the masked phrase,*"the chair"*, and the other candidate is a different phrase in the sentence, *"the cup"* extracted from *"She put the cup on the chair"*. The constraint we impose is that both the ground-truth noun-phrase and the alternative candidate need to appear before the masked phrase location, which mimics the pronoun resolution task. We sample sentences following the above constraint to create our synthetic datasets for pre-training.

We convert the sentence into the format of {[CLS] *first_half* **option** *second_half* [SEP]} where *first_half* refers to *"She put the cup on the chair but he knocked over "* and *second_half* refers to *", and the cup fell."*. The **option** is replaced by candidates, *"the cup"* or *"the chair"*. We compute P(*the chair*|*sentence*, θ) and P(*the cup*|*sentence*, θ) and optimize θ, the parameters of the model, using cross-entropy loss. We use the final layer [CLS] vector from transformer-based language models and pass it through a single layer feed-forward network to calculate the logits.

3.1 Discussion

The intuition behind MNPP is that given sufficient samples that mimic pronoun resolution task, the model can learn rich knowledge to perform well on human-annotated pronoun resolution datasets. Such idea is also in-line with recent advances in unsupervised QA (Lewis et al., 2019; Li et al., 2020; Banerjee and Baral, 2020; Banerjee et al., 2020, 2021), where synthetic QA datasets are created from unannotated corpora to perform unsupervised pre-training. Strictly speaking, MNPP is even more unsupervised since our synthetic datasets are not created with true pronoun resolution signals, whereas synthetic QA datasets in works cited above contain true question-answer pairs.

As mentioned in previous Section 2, similar to our work, Kocijan et al. (2019b) studied such pre-training strategy by constructing a synthetic dataset, called MaskedWiki, which is crawled from English Wikipedia. However, our work is significantly different from theirs in the following ways. First, their

Dataset \ Source	CNN	QUOREF	Gutenberg	Knowledge	Total
Hybrid Source	100,556	51,451	6,381	-	158,388
Hybrid Source w/ Knowledge	189,376	98,844	19,424	75,993	383,637

Table 2: Number of instances from each source of two hybrid-source synthetic datasets in the first group.

Synth. Dataset \ Downstream	WinoGrande (AUC)	WSC	DPR	KnowRef	COPA
Hybrid Source (160k)	58.08 (**0.6961**)	**79.48**	82.27	79.83	71.29
Hybrid Source w/ Know. (380k)	58.56 (0.6821)	78.39	**83.88**	79.04	73.27
Gutenberg-10k	57.93 (-)	75.09	81.21	77.15	79.21
Gutenberg-50k	57.40 (-)	76.19	77.84	75.10	74.26
Gutenberg-100k	58.56 (-)	72.53	75.00	74.40	75.25
Gutenberg-300k	57.38 (-)	75.82	81.56	76.44	78.22
Gutenberg-500k	**59.19** (0.6748)	76.56	80.50	79.12	**85.51**
Gutenberg-Easy (33k)	56.43 (-)	69.60	70.92	75.10	77.23
Gutenberg-Medium (33k)	57.00 (-)	75.10	80.32	78.17	79.21
Gutenberg-Hard (33k)	57.54 (-)	75.82	80.67	**79.98**	74.36

Table 3: Zero-shot transfer performances (%) on downstream datasets. AUC scores of WinoGrande are calculated after finetuning on all 5 splits of WinoGrande training sets. Difficulty level is decided using cosine similarity between the two candidate word vectors. Hard samples are the top 33% of samples when they are sorted in descending order using similarity score. Easy are bottom 33%, with Medium in-between.

pipeline requires further finetuning on another pronoun resolution task before transferring to downstream datasets, whereas our method can be directly evaluated on downstream datasets. Second, the size of MaskedWiki is 2.4 millions, which is 15 times the size of our best performing synthetic dataset. Third, we study how different properties of synthetic datasets affect zero-shot performances. Finally, they use a masked token prediction loss, and we model it as a classification task. Kocijan et al. (2019a) also construct another synthetic dataset called WikiCREM following the same masking principle but with only personal names masked.

4 Experiments and Results

4.1 Synthetic Dataset

We study three properties of synthetic dataset: source style, size, and difficulty level. The sources we choose include various styles of texts, including CNN stories (See et al., 2017), Wikipedia, and PG-19 language modeling benchmark (Rae et al., 2020). We study 3 groups and a total of 10 different synthetic datasets. The first group contains two synthetic datasets collected from all sources with and without knowledge hunting strategy (Prakash et al., 2019). The second group contains five synthetic datasets collected only from PG-19 but with varying sizes from 10k to 500k. The third group contains three synthetic datasets collected from PG-19 but with easy, medium, and hard samples with the same size of 33k each.[3] Datasets' names are listed in the first column of Table 3 and statistics of the first group are described in Table 2.

4.2 Unsupervised Pronoun Resolution

The downstream datasets we test on are the WinoGrande test set (17k instances), DPR test set (564 instances), KnowRef test set (12k instances), and COPA validation set (101 instances). Although COPA (Wang et al., 2019a) is a cause and effect identification dataset, Sakaguchi et al. (2020) show that directly transferring from a WinoGrande-finetuned RoBERTa-large model to COPA already achieves a good performance, indicating that finetuning on WinoGrande can serve as a resource for common sense knowledge. We also investigate whether learning through MNPP can serve as a resource for common sense. Note that we also provide evaluation on the GAP dataset (Webster et al., 2018) in Table 5 for reference although the authors of GAP explicitly mention in their paper that they urge the community to not treat GAP as a Winograd-style task but a co-reference resolution task without gold mention provided.

4.2.1 Results

We report our experiment results in Table 3 and Table 4. Table 3 shows that different downstream

[3]Please refer to supplemental materials for details on synthetic datasets constructions.

WSC (Levesque et al., 2012)	
Bi-LSTM-DPR (2018)	56.0
BERT_NSP-DPR (2019)	71.1
CorefBERT$_{\text{LARGE}}$ (2020)	71.4
BERT-WIKICREM-DPR (2019a)	71.8
BERT-MASKEDWIKI-DPR (2019b)	72.5
UDSSM-MASKEDWIKI-DPR (2019)	75.1
AMS-CSQA-DPR (2019)	75.5
RoBERTa-DPR (2020)	83.1
CorefRoBERTa$_{\text{LARGE}}$ (Ye et al., 2020)	83.2
RoBERTa-WG (2020)	**90.1**
Modified Skip-Gram (2018)	60.3
BERT Inner Attention (2019)	60.3
BERT-MASKEDWIKI (2019b)	61.9
UDSSM (2019b)	62.4
BERT-WIKICREAM (2019a)	63.4
Ensemble LMs (2018)	63.7
CSS (2020)	69.6
GPT-2 (2019)	70.7
WSC Know. Hunting (2019)	71.1
MNPP (this work)	**79.5**

WinoGrande (Sakaguchi et al., 2020)		**AUC**
RoBERTa (local context) (2020)	50.0	-
BERT-DPR (2020)	51.0	-
BERT (local context) (2020)	51.9	-
RoBERTa-DPR (2020)	58.9	-
BERT (2020)	64.9	0.5289
CSS (2020)	65.0	0.6046
UnifiedQA-Bart-large (2020)	73.3	0.6358
CorefRoBERTa$_{\text{LARGE}}$ (2020)	77.9	-
RoBERTa-large (2020)	79.1	0.6641
CorefBERT$_{\text{LARGE}}$ (2020)	80.8	-
TTTTT (2020)	84.6	0.7673
UnifiedQA-T5-11B (2020)	**89.4**	0.8571
Wino Know. Hunting (2020)	49.6	-
Ensemble LMs (2020)	50.9	-
MNPP (this work)	**59.2**	0.6706

DPR (Rahman and Ng, 2012)	
Bi-LSTM (2018)	63.0
FeatureEng+Ranking (2012)	73.0
BERT-WIKICREM-DPR (2019a)	80.0
BERT-DPR (2019a)	83.3
BERT-MASKEDWIKI-DPR (2019b)	84.8
BERT-WG (2020)	84.9
CorefBERT$_{\text{LARGE}}$ (Ye et al., 2020)	85.1
RoBERTa-DPR (2020)	91.7
CorefRoBERTa$_{\text{LARGE}}$ (Ye et al., 2020)	92.2
RoBERTa-WG (2020)	92.5
RoBERTa-WG-DPR (2020)	**93.1**
BERT-WIKICREAM (2019a)	67.4
CSS (2020)	80.1
MNPP (this work)	**83.9**
KnowRef (Emami et al., 2019)	
E2E-CoNLL (2019)	60.0
E2E-KnowRef (2019)	61.0
BERT (2019)	65.0
E2E-KnowRef+CoNLL (2019)	65.0
RoBERTa-DPR (2020)	84.2
RoBERTa-WG (2020)	**85.6**
CSS (2020)	65.5
MNPP (this work)	**80.0**
COPA (Wang et al., 2019a)	
RoBERTa-WG (2020)	84.4
MNPP (this work)	**85.5**

Table 4: Comparisons of zero-shot transfer performance (%) among baselines and MNPP. Works highlighted with gray are supervised methods either directly finetuned on downstream datasets or additionally finetuned on another pronoun resolution dataset. Works highlighted with cyan are fully unsupervised methods. Best performances are in bold. We also underline supervised methods that our method outperforms. Note that AUC score for MNPP is obtained after finetuning on all WinoGrande data splits. (Model-A-B stands for model finetuned on A and B sequentially.)

dataset benefits from different property of the synthetic dataset. The hybrid-source synthetic dataset of size 160k outperforms PG-500k by a large margin on both WSC and DPR. It shows that pre-training on text of various styles instead of larger size is probably a better guarantee for better zero-shot performance on WSC and DPR. However, on WinoGrande and KnowRef, text style and dataset size both seem to impact zero-shot performance. On WinoGrande, larger size matters slightly more, whereas on KnowRef, synthetic dataset with various styles of texts gives better performance. On COPA, it is clear that using books as the source and with larger size at the same time is the key, probably because fictional event descriptions describing day-to-day activities in books contain more common sense, whereas CNN or Wikipedia articles contain precise, factual, non-fictional event descriptions. Finally, pre-training on more challenging examples helps on all tasks except COPA.

Compared with previous methods in Table 4, MNPP outperforms all unsupervised methods on all datasets and is comparable with several strong supervised methods. Current best unsupervised methods on WinoGrande is either random guess or below it, however, MNPP outperforms all of them by a margin of at least 8%. Even compared with a supervised baseline where BERT is first finetuned on DPR, our method outperforms it by 8%. On WSC, MNPP also outperforms all SOTA unsupervised methods by more than 8% and outperforms most supervised methods by at least 4% except RoBERTa-large finetuned on another pronoun resolution dataset. On DPR, our method outperforms the SOTA unsupervised baseline over 3% and also achieves only 1% behind the strong supervised baseline that finetunes BERT on MaskedWiki and DPR sequentially or only on WinoGrande. On KnowRef, MNPP outperforms the only unsuper-

	M	F	B	O
BERT (Kocijan et al., 2019a)	75.3	75.1	1.00	75.2
CorefBERT$_{\text{LARGE}}$ (Ye et al., 2020)	-	-	-	76.8
BERT-WIKICREM-GAP (Kocijan et al., 2019a)	76.4	78.4	1.03	77.4
CorefRoBERTa$_{\text{LARGE}}$ (Ye et al., 2020)	-	-	-	77.8
BERT-WIKICREM-ALL-GAP (Kocijan et al., 2019a)	76.7	79.4	1.04	**78.0**
BERT-WIKICREM (Kocijan et al., 2019a)	60.5	57.5	0.95	59.0
MNPP (this work)	71.3	75.2	1.05	**73.3**

Table 5: Performance comparisons among previous works and MNPP on GAP measured in F1. M stands for male, F stands for female, B stands for bias, and O stands for overall. Works highlighted with lightgray are supervised methods and works highlighted with cyan are fully un-supervised methods.

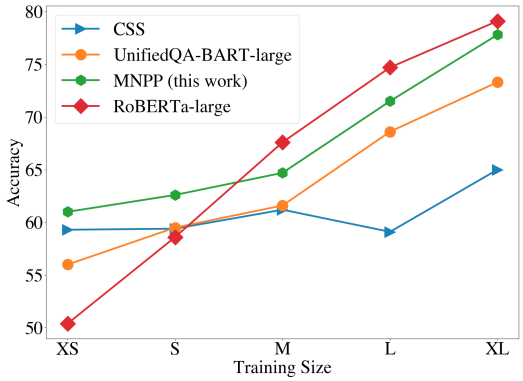


Figure 1: Performances (%) on WinoGrande test set after finetuning on 5 sizes of WinoGrande training set.

vised baseline by nearly 15% and achieves only 5% behind SOTA supervised model. Finally, on COPA, we show that MNPP gives models better common sense knowledge than finetuning on WinoGrande.

Meanwhile, we are not surprised that SOTA supervised methods still outperform unsupervised methods, including ours, considering the supervision itself and huge models with billions of parameters such as T5-11B.

4.3 Few-Shot Pronoun Resolution

We further proceed to the few-shot setting on WinoGrande-S and XS. We take the top three performance zero-shot models on WinoGrande development set and finetune them on WinoGrande-XS (160 instances) and S (640 instances) separately. After few-shot evaluation, we also finetune on the remaining three data splits, which are WinoGrande-M, L, and XL. Best performances on all 5 data splits are reported in Fig. 1 and AUC scores are reported in thrid column of WinoGrande section in Table 4.

4.3.1 Results

As indicated in Figure 1, MNPP outperforms CCS, UnifiedQA-BART-large, and RoBERTa-large on WinoGrande-S and XS with a large margin, and more importantly, achieves a higher AUC score as indicated in Table 4. It is clear that MNPP pretraining gives the model crucial additional information in the few-shot setting where only minimal data is available. We also notice that in the AUC column of Table 3, there is a negative correlation between zero-shot performance and AUC score, which means higher zero-shot performance does not guarantee better finetuning results.

Again we need to mention that we are not comparing with SOTA performances from billions-parameters models such as UnifiedQA-T5-11B from Khashabi et al. (2020) or T5-3B from Lin et al. (2020).

5 Conclusion

In this work, we propose MNPP pre-training to tackle unsupervised pronoun resolution and study how different properties of the synthetic pre-training dataset impact zero-shot performance on downstream datasets. Without finetuning on any pronoun resolution signal, MNPP outperforms all previous fully unsupervised methods on all tasks we study and even several strong supervised baselines. In the few-shot case where we finetune the zero-shot transfer model on WinoGrande-S and XS respectively, our model outperforms baselines by large margins, and further achieves a higher AUC score.

This work shows the effectiveness of unsupervised task definitions on text-based pronoun-resolution and common sense reasoning tasks. It would be interesting to design such tasks for multi-modal common sense reasoning (Zellers et al., 2019; Fang et al., 2020).

Acknowledgements

The authors acknowledge support from the DARPA SAIL-ON program W911NF2020006, ONR award N00014-20-1-2332, and NSF grant 1816039; and thank Yulong Chen for proofreading and the anonymous reviewers for their insightful discussion.

References

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162, Online. Association for Computational Linguistics.

Pratyay Banerjee, Tejas Gokhale, and Chitta Baral. 2021. Self-supervised test-time learning for reading comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1200–1211, Online. Association for Computational Linguistics.

Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. 2020. Self-supervised vqa: Answering visual questions using images and captions. *arXiv preprint arXiv:2012.02356*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ali Emami, Paul Trichelair, Adam Trischler, Kaheer Suleman, Hannes Schulz, and Jackie Chi Kit Cheung. 2019. The KnowRef coreference corpus: Removing gender and number cues for difficult pronominal anaphora resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3952–3961, Florence, Italy. Association for Computational Linguistics.

Zhiyuan Fang, Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020. Video2Commonsense: Generating commonsense descriptions to enrich video captioning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 840–860, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. 2019. A hybrid neural network model for commonsense reasoning. *arXiv preprint arXiv:1907.11983*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Tassilo Klein and Moin Nabi. 2019. Attention is (not) all you need for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4831–4836, Florence, Italy. Association for Computational Linguistics.

Tassilo Klein and Moin Nabi. 2020. Contrastive self-supervised learning for commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7517–7523, Online. Association for Computational Linguistics.

Vid Kocijan, Oana-Maria Camburu, Ana-Maria Cretu, Yordan Yordanov, Phil Blunsom, and Thomas Lukasiewicz. 2019a. WikiCREM: A large unsupervised corpus for coreference resolution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4303–4312, Hong Kong, China. Association for Computational Linguistics.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019b. A surprisingly robust trick for the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,

pages 4837–4842, Florence, Italy. Association for Computational Linguistics.

Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, and Leora Morgenstern. 2020. A review of winograd schema challenge datasets and approaches. *arXiv preprint arXiv:2004.13831*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*. Citeseer.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and refining question-answer pairs for unsupervised QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728, Online. Association for Computational Linguistics.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Tttttackling winogrande schemas. *arXiv preprint arXiv:2003.08380*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Leora Morgenstern, Ernest Davis, and Charles L Ortiz. 2016. Planning, executing, and evaluating the winograd schema challenge. *AI Magazine*, 37(1):50–54.

Juri Opitz and Anette Frank. 2018. Addressing the Winograd schema challenge as a sequence ranking task. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 41–52, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Ashok Prakash, Arpit Sharma, Arindam Mitra, and Chitta Baral. 2019. Combining knowledge hunting and neural language models to solve the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6110–6119, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Yu-Ping Ruan, Xiaodan Zhu, Zhen-Hua Ling, Zhan Shi, Quan Liu, and Si Wei. 2019. Exploring unsupervised pretraining and sentence structure modelling for winograd schema challenge. *arXiv preprint arXiv:1904.09705*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Shuohang Wang, Sheng Zhang, Yelong Shen, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Jing Jiang. 2019b. Unsupervised deep structured semantic models for commonsense reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 882–891, Minneapolis, Minnesota. Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv preprint arXiv:1908.06725*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6724.

Hongming Zhang and Yangqiu Song. 2018. A distributed solution for winograd schema challenge. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 322–326.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A Related Work on Supervised Methods

WSC & DPR. Opitz and Frank (2018) is the first work to propose transfer learning from another pronoun resolution dataset such as DPR to WSC. He et al. (2019) use a hybrid model of Wang et al. (2019b) and Kocijan et al. (2019b). Ruan et al. (2019) explore BERT's next sentence prediction with finetuning on DPR. Ye et al. (2020) finetune a new language representation model called CorefBERT, which is trained with a novel task to strengthen the co-referential reasoning ability of BERT, on DPR and then test on DPR and WSC. The SOTA supervised performance is provided by Sakaguchi et al. (2020) where they finetune a RoBERTa-large model on WinoGrande or DPR and evaluate on WSC and DPR without and with further finetuning. A detailed review of WSC and WSC related dataset can be found at Kocijan et al. (2020).

KnowRef. In Emami et al. (2019), an end-to-end neural system (Lee et al., 2018) is trained on CoNLL2012 shared task (Pradhan et al., 2012) and then tested under three settings: directly applying to KnowRef test set, retraining on KnowRef, and retraining on KnowRef plus CoNLL2012. Sakaguchi et al. (2020) transfer a WinoGrande-finetuned RoBERTa-large model and DPR-finetuned RoBERTa-large model to KnowRef test set respectively.

WinoGrande. The authors of WinoGrande finetune a RoBERTa-large on WinoGrande training set and evaluate on the test set in standard supervised setting, and Lin et al. (2020) finetune a T5-3B model instead. Sakaguchi et al. (2020) also study finetuning BERT and RoBERTa with only local context (only tokens near the pronoun location are available instead of the whole sentence). Ye et al. (2020) finetune WinoGrande using CorefBERT. Klein and Nabi (2020) finetune their unsupervised CSS model. Finally, UnifiedQA (Khashabi et al., 2020), which is pre-trained on eight seed QA datasets spanning four different formats in a unified way, is finetuned on WinoGrande.

B Synthetic Datasets Construction

For the first synthetic dataset in the first group, we choose 5000 stories in CNN stories, a small portion of Gutenberg books, and the whole training set of QUOREF (Dasigi et al., 2019), which is a reading comprehension dataset that requires resolving coreference among entities crawled from Wikipedia, and these sources result in the size of 160k. The second synthetic dataset in the first group comprises the same sources as above plus extra knowledge crawled by Google query using the knowledge hunting strategy introduced in Prakash et al. (2019). Following their strategy, we scrap 6531 and 69462 knowledge sentences for WSC and WinoGrande respectively. We relax the filtering process to allow longer sentences than those in the first synthetic dataset and lead to 380k samples in total. We then fix the text style and study the influence of data size on pre-training. We use 2000 books from PG-19 as the source and create five synthetic datasets with size of 500k, 300k, 100k, 50k, and 10k as the second group. We further study how difficulty levels of samples affect the downstream zero-shot performance. We select 100k samples from the PG-19 books described above and evenly split them into three synthetic datasets with low, medium, and high similarity scores between candidate choices as the third group. As a result, we create 3 groups of synthetic datasets with ten synthetic datasets in total. We used spaCy[4] to pre-process raw text, including removing blank spaces, special characters, sentences that are too short or too long, and extracting noun-phrases.

C Zero-shot Experiment Details

Recent study (Khot et al., 2020) has shown that finetuning a RACE-finetuned (Lai et al., 2017) RoBERTa model as a start point is much more stable than directly finetuning a RoBERTa model from scratch, we follow the same strategy to start finetuning a RACE-finetuned RoBERTa-large model on all synthetic datasets. We use Hugging Face Transformers[5] as our codebase. We set Adam optimizer with an initial learning rate of $1e-5$ and epsilon of $1e-8$, and without weight decaying for all settings. For a synthetic dataset whose size is larger or equal to 100k, we choose the batch size of 32 and train for 20 epochs, otherwise, we choose the batch size of 16 and train for 50 epochs. We checkpoint every X steps, with X in [50,500].

D Few-shot Experiment Details

We set Adam optimizer with an initial learning rate of $1e-5$ and epsilon of $1e-8$, without weight decaying, and batch size between 16 and 32 for all

[4] https://spacy.io/
[5] https://github.com/huggingface/

sizes. We finetune 20 epochs for WinoGrande-XL, L, and M, 40 epochs for S, and 160 epochs for XS. We checkpoint every X steps, with X in [50,500].

Addressing Semantic Drift in Generative Question Answering with Auxiliary Extraction

Chenliang Li, Bin Bi, Ming Yan
Wei Wang, Songfang Huang
Alibaba Group
{lcl193798, b.bi, ym119608}@alibaba-inc.com
{hebian.ww, songfang.hsf}@alibaba-inc.com

Abstract

Recently, question answering (QA) based on machine reading comprehension has become popular. This work focuses on generative QA which aims to generate an abstractive answer to a given question instead of extracting an answer span from a provided passage. Generative QA often suffers from two critical problems: (1) summarizing content irrelevant to a given question, (2) drifting away from a correct answer during generation.

In this paper, we address these problems by a novel Rationale-Enriched Answer Generator (REAG), which incorporates an extractive mechanism into a generative model. Specifically, we add an extraction task on the encoder to obtain the rationale for an answer, which is the most relevant piece of text in an input document to a given question. Based on the extracted rationale and original input, the decoder is expected to generate an answer with high confidence. We jointly train REAG on the MS MARCO QA+NLG task and the experimental results show that REAG improves the quality and semantic accuracy of answers over baseline models.

1 Introduction

Question Answering (QA) has come a long way from answer sentence selection, relationship QA to machine reading comprehension (MRC). Recently, QA has become an essential problem in natural language understanding and a major milestone towards human-level machine intelligence. Current mainstream approaches (Chen et al., 2017; Wang et al., 2018; Yan et al., 2018) treat MRC as a process of extracting a consecutive piece of text from a document to a given question.

Despite the great success in extractive MRC (Wang et al., 2018; Chen et al., 2020), in real-world applications, correct answers may span different passages or even not be literally present in the passages. Directly extracting a consecutive answer span is often inadequate. Therefore, the ability of generating an abstractive answer is needed, which requires a QA model to summarize the main content in a paragraph that is relevant to a given question.

Question	does gameplay programmer need math skill
Passage	A good computer programmer is more of a problem solver and logical thinker than a math buff. Besides, the industry is peppered with many computer programmers who do not really know much about mathematics.
Gold	no, gameplay programmer does not need math skill.
PALM	yes, gameplay programmer is a math buff.
REAG	no, gameplay programmer does not need math skill.

Table 1: An example of the "semantic drift" issue in generative reading comprehension from the MARCO dataset (Nguyen et al., 2016). The text span of words in blue is the rationale extracted by REAG.

Answering questions in natural language can be beneficial to a variety of QA applications, and has led to the development of smart devices such as Siri, Cortana and Alexa. However, compared with answer extraction, answer generation for reading comprehension is more challenging, and has been less explored. A major challenge in generative reading comprehension comes from out-of-control generation of abstractive answers. Although much work has been done in neural language generation (NLG), e.g., KIGN(Li et al., 2018) for summarization, out-of-control generation remains an open question for generative QA which aims to produce correct and coherent answers. Specifically, we observed that generative models often generate answers semantically drifting away from the given

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 942–947
August 1–6, 2021. ©2021 Association for Computational Linguistics

passage and question, known as the "semantic drift" problem. As shown in Table 1, the baseline generative model PALM (Bi et al., 2020) generates an answer that has almost contrary semantics with the gold answer. In general, a generative model often suffers from two critical problems: (1) summarizing content irrelevant to a given question, and (2) drifting away from a correct answer during generation.

In this paper, we address these problems by a novel Rationale-Enriched Answer Generator (REAG), which incorporates an extractive mechanism into a generative model in order to leverage relevant information to a given question in the contextual passage. Specifically, we add an extraction task on the encoder to obtain the rationale for an answer, which is the most relevant piece of text in an input document to the given question. On one hand, the introduction of the supervised extraction task enables the encoder to learn the relevance between a question and a passage; On the other hand, the extracted rationale can be further used to guide the answer generation. Based on the extracted rationale and original input, the decoder is expected to summarize content relevant to a given question and generates an answer with high confidence. Finally, we jointly train REAG on the MS MARCO QA+NLG task based on the common bottom layers. The experimental results show that REAG improves the semantic accuracy of answers over the other state-of-the-art models.

2 Related Work

2.1 Machine Reading Comprehension

In recent years, machine reading comprehension has made great progress with the development of SQuAD (Rajpurkar et al., 2016) and MS MARCO (Nguyen et al., 2016). The current mainstream studies treat machine reading comprehension as answer span extraction from one passage (Rajpurkar et al., 2016, 2018) or multi-passages (Nguyen et al., 2016), which is usually done by predicting the start and end position of an answer. SLQA (Wang et al., 2018) improved answer quality with a hierarchical attention fusion network, which conducted attention and fusion horizontally and vertically across layers between a passage and a question. Recently, the BERT model Devlin et al. (2019) has proved effective for reading comprehension via unsupervised pre-training.

2.2 Generative Reading Comprehension

Bi et al. (2019) proposed a Knowledge-Enriched Answer Generator (KEAG) to compose a natural answer by exploiting and aggregating evidence from all four information sources available: question, passage, vocabulary and knowledge. Nishida et al. (2019a) proposed a multi-style generative model to generate an abstractive summary from the given question, passages and multi-style.

2.3 Reliable Text Generation

Compared with answer extraction, answer generation for reading comprehension is more challenging, and the major challenge in generative reading comprehension lies in out-of-control generation. Recently, some studies have been carried out on increasing the reliability of generation in the encoder-decoder framework(Liu et al., 2018; Li et al., 2018).

3 Rationale-Enriched Answer Generation

3.1 Rationale Span Extraction

In a generative reading comprehension task, every answer has its corresponding rationale, an extractive span in the passage, which can be derived by matching the passage text with the answer. The rationale can usually be located in a certain continuous area of the passage. We use continuous text span as the rationale to minimize the difficulty of the extraction task. Compared with the gold answer, the text span with the highest F1-score in passage is identified as the rationale for training supervision.

Based on the identified rationale, we introduce a rationale extraction task into the encoder. It enables the encoder to learn the relevance between the input question and the passage. Specifically, the encoder predicts whether each token of the passage should be included in the rationale. Every token in the rationale is labeled by 1 and the rest is labeled by 0.

Given input question Q and passage P, we first concatenate them together into an input sequence $X = \{x_1, x_2, ..., x_N\}$. Then we use a shared word embedding layer to project each of the vectors into d-dimensional vectors, and add to each the corresponding position embedding. The resulting vectors are then fed into the Transformer encoder to map the text into a sequence of encoder hidden states $\{h_1, h_2, ..., h_N\}$.

The encoder hidden states can be used to predict whether each token of the passage should be included in the rationale. Therefore, we add a fully connected layer with the sigmoid activation on top of the encoder, to compute the probability for each input word:

$$p_i^r = \text{sigmoid}(w_1 \cdot \text{relu}(W_2 h_i)) \quad (1)$$

where $h_i \in R^d$ is the output hidden state of the encoder for the i^{th} token.

This gives the probability p_i^r that the i^{th} token should be included in the rationale. We then calculate the averaged cross entropy, similar to(Ju et al., 2019), for the rationale extraction loss:

$$\mathcal{L}_{RE_j} = -\frac{1}{N}\sum_{i=1}^{N}(y_{ji}^r \log p_{ji}^r + (1-y_{ji}^r)\log(1-p_{ji}^r)), \quad (2)$$

where N is the number of input tokens. y_i^r is the rationale label for the i^{th} token, and $\mathcal{L}_{RE_j}$ represents the rationale loss for the j^{th} example in the training set.

3.2 Rationale-Enriched Answer Generation

This layer uses a stack of Transformer decoder blocks on top of the embeddings provided by the encoder's word embedding layer. The decoder is similar in structure to the encoder except that it includes a standard attention mechanism after each self-attention layer that attends to the output of the encoder. The rationale-aware hidden states output by the encoder are used for rationale extraction.

In calculating the decoder states s_t, an cross attention is introduced into the decoder to attend to the rationale-aware encoder hidden states. This results in the rationale-aware decoder hidden state s_t:

$$p(y_t|y_1, ..., y_{t-1}) = \text{softmax}(W^e(W^v s_t + b^v) + b^e) \quad (3)$$

During training, we minimize the negative log-likelihood of the answer word at each decoding time step. Let y_t^* denote the target word in the decoding time step t. The overall loss is then defined as:

$$\mathcal{L}_{GEN} = -\frac{1}{T}\sum_{t=1}^{T}\log p(y_t^*|y_1^*, ..., y_{t-1}^*, x, \theta) \quad (4)$$

where T denotes the length of a gold answer.

3.3 Joint Training and Prediction

The rationale extraction task and the answer generation task are designed to share the same embedding and the encoder. Therefore, we propose to train them together as multi-task learning. The joint objective function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{GEN} + \beta\mathcal{L}_{RE} \quad (5)$$

where β is a hyper-parameter that controls the weight of the rationale extraction task. During the training process, we use a linear decay schedule on the value of β, in order to rely more on the rationale extraction task for addressing the semantic drift problem at the early stage, following by more focus on the target generation task subsequently.

4 Experiments

4.1 Experiment Configuration

Dataset and Evaluation Metric. Given our objective of generating natural answers by document reading, the MARCO dataset [1] (Nguyen et al., 2016) released by Microsoft is a good fit for benchmarking REAG and other answer generation methods. We use the latest MARCO V2.1 dataset and focus on the "QA + Natural Language Generation" task in the evaluation. The data has been split into a training set (150k QA pairs), a dev set (12k QA pairs) and a test set (110k questions). Since true answers are not available in the test set and the task is retired now, we hold out the dev set for evaluation in our experiments, and test models for each question on its associated passages by concatenating them all together. Following Bi et al. (2019), we tune the hyper-parameters by cross-validation on the training set.

Implementation Details. Our REAG is based on PALM (Bi et al., 2020), an encoder-decoder generative language model pre-trained on a large corpus. It consists of a 12-layer encoder and 12-layer decoder with 768 embedding/hidden size, 3072 feed-forward filter size and 12 attention heads. REAG is trained with a dropout of 0.1 on all layers and attention weights. During training and testing, we truncate the text to 512 tokens and limit the length of the answer to 50 tokens. At test time, answers are generated using beam search with a beam size 5.

[1] https://microsoft.github.io/msmarco/

Model	ROUGE-L	BLEU-1
BIDAF+Seq2Seq[a]	34.15	29.68
S-Net[b]	42.71	36.19
S-Net+Seq2Seq[b]	46.83	39.74
gQA[c]	45.46	40.22
KEAG[d]	51.68	45.97
Masque[e]	69.77	65.56
PALM[f]	69.87	66.31
REAG	**70.98**	**69.12**

Table 2: Performance of generative reading comprehension in ROUGE-L and BLEU-1 on MARCO Q&A+NLG. All our ROUGE scores have a 95% confidence interval of at most ±0.25. [a](Seo et al., 2016);[b](Tan et al., 2017);[c](Mitra, 2017);[d](Bi et al., 2019);[e](Nishida et al., 2019b);[f](Bi et al., 2020).

Ablation	ROUGE-L	BLEU-1
REAG	**70.98**	**69.12**
✗ rationale-span extraction	69.87	66.31
✗ linear-decay joint training	70.45	68.28
✗ pre-training	69.54	68.12

Table 3: Ablation tests of REAG on the MARCO Q&A+NLG dataset.

4.2 Model Comparisons

Table 2 gives the comparison of other state-of-the-art QA models on the MARCO Q&A+NLG dataset in ROUGE-L and BLEU-1. From this table, we observe that generative QA models (e.g., REAG, PALM) are consistently superior to extractive models (e.g., BiDAF) in answer quality. Therefore, generative QA models establish a strong base architecture to be enhanced with the extra signals, which motivates this work. Among the generative models, REAG outperforms all the other state-of-the-art models with an improvement of over 2.8% BLEU-1 point and 1.1% ROUGE-L. Part of the results in the Table 2 are from (Bi et al., 2019), which re-running other researchers' code.

4.3 Ablation Study

We conduct ablation studies to assess the individual contribution of every component in REAG. Table 3 reports the results of full REAG and its ablations on the MS MARCO Q&A NLG dataset.

We evaluate how much rationale-span extraction contributes to generation quality by removing it from the REAG model. This ablation results in a drop from 70.98 to 69.87 on Rouge-L, demonstrating the role of the rationale-span extraction in REAG. In addition, we ablate the linear-decay joint-training which proves to be critical with over 0.5% drops on the metrics after the ablation. In order to exclude the influence of the pre-trained model, we ablate pre-training, retaining the rationale-span extraction. This ablation leads to a drop from 70.98 to 69.54 on Rouge-L, which demonstrates the power of REAG in generating high-quality answers without pre-training.

Method	Semantic Acc	ROUGE-L	BLEU-1
PALM	81.67	69.87	66.31
REAG	**84.33**	**70.31**	**68.59**

Table 4: Comparison of the semantic accuracy, ROUGE-L and BLEU-1 of REAG with those of PALM

	ROUGE-L	BLEU-1
Generated Answers	47.25	50.34
Gold Answers	38.14	43.12

Table 5: Agreement of generated/gold answers with extracted rationales for REAG

4.4 Quantitative Analysis on Semantic Drift

For generative reading comprehension, it is difficult to make the answer completely correct, because even if the semantics are correct, there may be some expression differences from the gold answer. Since neither ROUGE-L nor BLEU-1 can measure it, we conduct a human evaluation of the semantic accuracy. We randomly select 100 questions from the MARCO dev set, and manually evaluate whether the generated answers to these questions are semantically drifted. Table 4 reports the semantic accuracy of REAG and PALM obtained by human. Our REAG model surpasses PALM in generating correct answers without semantic drift. Although our REAG model improves over PALM by 1.1% in automatic evaluation metric ROUGE-L, it gives a 3.26% improvement in semantic accuracy. This shows the fact that in some cases automatic evaluation metrics, such as ROUGE-L and BLEU-1, do not reflect semantic accuracy.

In addition, we compute the agreement of generated/gold answers with extracted rationales for

Example 1	
Relevant Passage	Yes | No Thank you! Flu shots are not made for children under the age of 6 months. If you read the vaccine insert and studies regarding the flu shot and kids, you will see that flu shots don't even work for children under the age of 2.
Question	can a child get a flu vaccine under 6 months?
Gold Answer	No, a child under 6 months can't be given a flu vaccine.
PALM Answer	Yes, a child can get a flu vaccine under 6 months.
REAG Answer	No, a child cannot get a flu vaccine under 6 months.

Example 2	
Relevant Passage	Modesto, Stanislaus County Sales Tax Rate. Details. The sales tax in Modesto is 7.625%, which is about average for cities in Stanislaus County and lower than average for California (8%). Modesto is one of 21 cities in Stanislaus County with a distinct sales tax as listed by the California Board of Equalization. See all cities in Stanislaus County. Advertisement.
Question	what is the sales tax in california
Gold Answer	The sales tax in California is 8%.
PALM Answer	The sales tax in California is 7.625%
REAG Answer	The sales tax in California is 8%

Table 6: Examples of the output of REAG and PALM on the MARCO dataset. The text span of words in blue is the rationale extracted by REAG

REAG in ROUGE-L and BLEU-1. As shown in Table 5, the generated answers are strongly correlated with the rationales, demonstrating the effectiveness of leveraging the rationale signal. Also, the fact that the gold answers have a lower agreement with the rationales indicates that a generative model, as opposed to an extractive one, is needed for the MARCO Q&A+NLG task.

4.5 Case Study

Table 6 gives two examples to show the answers generated by the REAG model and the PALM model. In addition to the answers, we provide the rationales predicted by REAG's encoder to demonstrate the effectiveness of rationale extraction. In both examples, the rationale extraction module identifies the correct rationales, e.g., *Flu shots are not made for children under the age of 6 months.* and *California (8%)*.

In Example 1, PALM is confused by the noise "*Yes*" in the beginning of the passage, which leads to the contrary semantics of its generated answer. With the correctly extracted rationale, our REAG model generates an answer semantically consistent with the gold answer. In Example 2, PALM fails to identify a correct sales tax rate *8%* for California, so the response is incorrect and useless, even if it results in high ROUGE and BLEU scores against the gold answer. In contrast, based on the extracted rationale *California (8%)*, our REAG generates a semantically correct answer.

5 Conclusion and Future Work

This paper presents a novel model REAG that is designed to incorporate an extractive mechanism into a generative QA model. REAG introduces a new task on the encoder to extract rationales. Based on these rationales and original input, a rationale-enriched decoder is proposed to generate an answer with high confidence. The experimental results show that REAG significantly improves the quality and semantic accuracy of generated answers over state-of-the-art models.

References

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, and Wei Wang. 2020. Palm: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2521–2530, Hong Kong, China. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020. Question directed graph attention network for numerical reasoning over text. *arXiv preprint arXiv:2009.07448*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on Conversational Question Answering. *arXiv e-prints*, page arXiv:1909.10772.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, New Orleans, Louisiana. Association for Computational Linguistics.

Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119, Brussels, Belgium. Association for Computational Linguistics.

Rajarshee Mitra. 2017. A Generative Approach to Question Answering. *arXiv e-prints*, page arXiv:1711.06238.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, and Rangan Majumder. 2016. Ms marco: A human generated machine reading comprehension dataset.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019a. Multi-style generative reading comprehension. *CoRR*, abs/1901.02262.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019b. Multi-style generative reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2273–2284, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Chuanqi Tan, Furu Wei, Nan Yang, Weifeng Lv, and Ming Zhou. 2017. S-net: From answer extraction to answer generation for machine reading comprehension. *CoRR*, abs/1706.04815.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714, Melbourne, Australia. Association for Computational Linguistics.

Ming Yan, Jiangnan Xia, Chen Wu, Bin Bi, Zhongzhou Zhao, Ji Zhang, Luo Si, Rui Wang, Wei Wang, and Haiqing Chen. 2018. A deep cascade model for multi-document reading comprehension. In *AAAI*.

Demoting the Lead Bias in News Summarization via Alternating Adversarial Learning

Linzi Xing*, **Wen Xiao***, **Giuseppe Carenini**
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
{lzxing, xiaowen3, carenini}@cs.ubc.ca

Abstract

In news articles the lead bias is a common phenomenon that usually dominates the learning signals for neural extractive summarizers, severely limiting their performance on data with different or even no bias. In this paper, we introduce a novel technique[1] to demote lead bias and make the summarizer focus more on the content semantics. Experiments on two news corpora with different degrees of lead bias show that our method can effectively demote the model's learned lead bias and improve its generality on out-of-distribution data, with little to no performance loss on in-distribution data.

1 Introduction

Neural extractive summarization, which produces a short summary for a document by selecting a set of representative sentences, has shown great potential in real-world applications, including news (Cheng and Lapata, 2016; Nallapati et al., 2017) and scientific paper summarization (Cohan et al., 2018; Xiao and Carenini, 2019). Typically, a general-purpose extractive summarizer learns to select the most important sentences from a document to form the summary by considering their content salience, informativeness and redundancy. However, when restricted to a specific domain, the summarizer can learn to exploit particular biases in the data, the most famous of which is the *lead bias* in news (Nenkova et al., 2011; Hong and Nenkova, 2014); namely that sentences at the beginning of a news article are more likely to contain summary-worthy information. As a result, not surprisingly, such bias is strongly captured by neural extractive summarizers for news, for which the sentence positional information tends to dominate the actual content of the sentence in model prediction (Jung et al., 2019; Grenander et al., 2019; Zhong et al., 2019a,b).

While learning a summarizer reflecting the biases in the training dataset is completely fine when the summarizer is going to be deployed to summarize documents having similar biases, it would be problematic when the model was applied to deal with documents coming from a mixture of datasets with different degrees of such biases. In this paper, we address this problem in the context of the lead bias in the news domain by exploring ways in which an extractive summarizer for news can be trained so that it learns to balance the lead bias with the content of the sentences, resulting in a model that can be applied more effectively when the target documents belong to news datasets in which the lead bias is present in rather different degrees.

Recently, Grenander et al. (2019) proposes two preliminary solutions. One is to pretrain the summarizer on an automatic generated "unbiased" corpus where the document sentences are randomly shuffled, which however has the negative effects of preventing the learning of inter-sentential information. The other, which can be only applied to RL-based summarizers, is to add an explicit auxiliary loss to directly balance position with content. Alternatively, Zhong et al. (2019b) and Wang et al. (2019) investigate strategies to train the summarizer on multiple news datasets with different degrees of lead bias, but this may still be problematic when we apply the trained summarizer to the documents with lead bias not covered in the training data. Outside the summarization area, methods have also been proposed to eliminate data biases for other NLP tasks like text classification or entailment (Kumar et al., 2019; Clark et al., 2019, 2020).

Inspired by Kumar et al. (2019), we have developed an alternating adversarial learning technique to demote the summarizer lead bias, but also maintain the performance on the in-distribution

* The first two authors contributed equally to this work.
[1] https://github.com/lxing532/Debiasing

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 948–954
August 1–6, 2021. ©2021 Association for Computational Linguistics

data. We introduce a position prediction component as an adversary, and optimize it along with the neural extractive summarizer in an alternating manner. Furthermore, in contrast with Grenander et al. (2019) and Wang et al. (2019), our proposal is model-independent and only requires one type of news dataset as training input.

In this paper, we apply our proposed method to a biased transformer-based extractive summarizer (Vaswani et al., 2017) trained on CNN/DM training set (Hermann et al., 2015) and conduct experiments on two test sets with different degrees of lead bias: CNN/DM and XSum (Narayan et al., 2018), for in-distribution and generality evaluation respectively. The experimental results indicate that our proposed "debiasing" method can effectively demote the lead bias learned by the neural news summarizer and improve its generalizability, while still mostly maintaining the model's performance on the data with a similar lead bias.

2 Proposed Debiasing Method

Our method aims to demote the lead bias learned by the summarizer and encourage it to select content based more on the semantics covered in sentences. As shown in Figure 1, our method comprises two components: one for *Summarization* (red) and the other for sentence *Position Prediction* (green).

2.1 Summarization Component

Following previous work, we formulate extractive summarization as a sequence labeling task (Xiao and Carenini, 2019, 2020; Xiao et al., 2020). For a document $d = \{s_1, s_2, ..., s_k\}$, each sentence will be assigned a score $\alpha \in [0, 1]$. The summary will then be formed with the highest scored sentences. We adopt a transformer-based model (Vaswani et al., 2017) as our basic "biased" summarization component (red in Fig. 1), as shown to be heavily impacted by the lead bias (Zhong et al., 2019a). This component contains a transformer-based encoder Enc_{θ_t} and a multilayer perceptron (MLP) decoder Dec_{θ_s}, parameterized by θ_t and θ_s respectively. We use the averaged word embedding from Glove as sentence embedding as suggested in Kedzie et al. (2018). We optimize this summarization system by minimizing the loss:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N} CE(\alpha_i, y_i) \qquad (1)$$
$$\alpha_i = Dec_{\theta_s}(Enc_{\theta_t}(s_i))$$

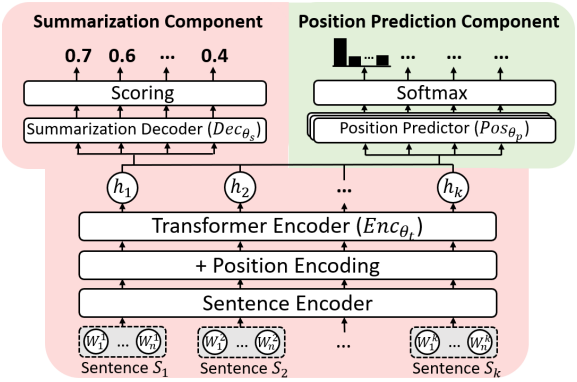


Figure 1: The overall architecture of our proposed lead bias demoting method.

where CE denotes the cross-entropy loss and $y_i \in \{0, 1\}$ is the ground truth label for sentence s_i, representing if s_i is selected to form the summary.

2.2 Position Prediction Component

Our goal is to train the summarization model to make accurate predictions based more on the sentence semantics, rather than whether the sentence is in the lead position. More specifically, we aim to design an encoder network Enc_{θ_t} to output the set of contextualized sentence representations $\mathbf{H} = \{h_1, ..., h_k\}$ which cover less sentence positional information, so that the following decoder Dec_{θ_s} will make predictions depending less on such positional information. To achieve this, the first step is to understand how much and in what form the positional information is encoded in Enc_{θ_t}. Therefore, we propose a position prediction network to learn to predict the position of sentences in a document based only on $\mathbf{H}$. Intuitively, the higher accuracy this component can achieve, the more positional information is contained in $\mathbf{H}$. This position prediction component will then play the role of an adversary module to demote the influence of lead bias presented in the training phase of the summarization component.

Concretely, because predicting the exact position for each sentence would require an extremely large set of labels with a skewed distribution, we choose to predict the portion of the document each sentence belongs to. In particular, once we obtain the set of contextualized sentence representations $\mathbf{H}$ from the encoder network Enc_{θ_t}, we initialize a MLP (parameterized by θ_p and followed by Softmax) as the position prediction component Pos_{θ_p} (green in Fig 1). In essence, this component Pos_{θ_p} takes $\mathbf{H}$ as input and outputs a M-dimensional multinomial distribution for each

sentence to represent its position in a document. More formally, $Pos_{\theta_p}(h_i) = (\hat{p}_1^{(i)}, .., \hat{p}_j^{(i)}, .., \hat{p}_M^{(i)})$ where $\sum_{j=1}^{M} \hat{p}_j^{(i)} = 1$. $\hat{p}_j^{(i)}$ is the predicted probability of the ith sentence belongs to the jth portion of a document when the document is divided into M parts (M is a tunable hyperparameter). We use the cross-entropy loss to optimize Pos_{θ_p} to extract sentence positional signals encoded in the system:

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N} CE(Pos_{\theta_p}(h_i), p_i) \qquad (2)$$

where p_i is the true position of sentence i.

2.3 Alternating Adversarial Learning

To demote the influence of positional bias and balance it with the sentence semantics in the summarization system, we want to modify the encoder to produce **H**, which can still be accurate for summary generation but fail at sentence position prediction. We achieve this by alternatingly executing "Position learning" and "Position debiasing", as proposed in Kumar et al. (2019) and presented in Algorithm 1. In the "Position learning" phase, once a pretrained summarization system is obtained, we first fix its weights and train an adversary network $Pos^*_{\theta_p}$ (sentence position predictor) to extract the positional information contained in the encoder. Then in the "Position debiasing" phase, we fix the weights of $Pos^*_{\theta_p}$ and update the parameters of the summarization component to maximize the position prediction loss of adversary (L_{adv} in eq 3) while minimizing the summarization loss L_1:

$$\begin{aligned} L_3 &= \beta L_1 + (1-\beta) L_{adv} \\ L_{adv} &= -\frac{1}{N}\sum_{i=1}^{N} CE(Pos^*_{\theta_p}(h_i), U_M) \end{aligned} \qquad (3)$$

To maximize the position prediction loss, the fixed adversary $Pos^*_{\theta_p}$ should ideally output the uniform distribution, $U_M = (\frac{1}{M}, ..., \frac{1}{M})$, for the position prediction of each sentence. β is the trade-off parameter tuned at validation stage to control the degree of lead bias demoting.

In practice, we notice that reusing the same adversary for all iterations will make the positional signals not weakened but instead encoded in a different way. To avoid this problem, we follow Kumar et al. (2019) to use multiple adversaries (parameterized with $[\theta_p^{(1)}, ..., \theta_p^{(N)}]$ in Algorithm 1), making it more difficult for the encoder to keep the sentence positional signals by encoding them into a more implicit format for position predictor to learn.

Algorithm 1: Alternating Adversarial Learning

Result: $\theta_t, \theta_s, [\theta_p^{(1)}, \cdots, \theta_p^{(N)}]$
Randomly initialize θ_t, θ_s;
while *not converged* **do**
  Sample a minibatch of training samples;
  Update θ_t, θ_s with gradients regarding L_1;
end — Pretraining
while $i <= N$ **do**
  for *p steps* **do**
    Randomly initialize $\theta_p^{(i)}$;
    Sample a minibatch of training samples;
    Fix θ_t, θ_s, update $\theta_p^{(i)}$ with gradients regarding L_2;
  end — Position Learning
  for *s steps* **do**
    Sample a minibatch of training samples;
    Randomly select $\theta_p^{(cur)}$ from $[\theta_p^{(1)}, \cdots, \theta_p^{(i)}]$;
    Fix $\theta_p^{(cur)}$, update θ_t and θ_s with gradients regarding L_3;
  end — Position Debiasing
  $i = i + 1$;
end

3 Experiments and Analysis

3.1 Datasets

We use the standard CNN/DM dataset (204,045 training, 11,332 validation and 11,334 test data) (Hermann et al., 2015) for training since it is one of the mainstream news datasets with observed lead bias (Jung et al., 2019; Grenander et al., 2019). For model evaluation, we use the test set of CNN/DM to evaluate model's in-distribution performance, as well as the test set of XSum (Narayan et al., 2018), which consists of 11,334 datapoints, to evaluate model's generality when transferred to less biased data. The empirical analysis in Narayan et al. (2018) and Jung et al. (2019) shows the documents and summaries in XSum are shorter and have less lead bias compared to CNN/DM.

3.2 Experimental Design

Baselines: We compare our proposal with various baselines (see Table 1). The top section of Table 1 presents ***Lead*** baseline and ***Oracle***. For CNN/DM, lead baseline refers to ***Lead-3*** and for XSum, it refers to ***Lead-1***. The middle section of Table 1 contains the basic transfomer-based summarizer accepting "sentence representation + position encoding" as input, and its two variants, one without positional encoding, while the other with only positional encoding as input. The bottom section contains ***Shuffling*** (Grenander et al., 2019), which

Model	CNN/DM				XSum			
	R1	R2	RL	Mean	R1	R2	RL	Mean
Lead	40.30	17.52	36.54	31.45	16.32	1.60	11.96	9.96
Oracle	56.04	33.10	52.29	47.14	30.98	8.98	23.51	21.16
Basic Transformer (Vaswani et al., 2017)	41.02	18.39	37.39	32.27	16.79	1.84	12.33	10.32
– No Position Encoding	37.82↓	15.59↓	34.32↓	29.24	**18.29**↑	**2.53**↑	**13.45**↑	**11.42**
– Only Position Encoding	40.13↓	17.36↓	36.38↓	31.29	16.22↓	1.62↓	11.90↓	9.91
Learned-Mixin (Clark et al., 2019)	40.72↓	18.27	37.17↓	32.05	16.67	1.91↑	12.28	10.29
Shuffling (Grenander et al., 2019)	**41.00**	**18.43**	**37.37**	**32.27**	16.98↑	1.96↑	12.48↑	10.47
Our Method	40.88↓⇓	18.37	37.27↓	32.18	17.20↑⇑	1.99↑⇑	12.63↑⇑	10.61

Table 1: The ROUGE-1/2/L F1 scores and "Mean" (mean of ROUGE-1/2/L) on CNN/DM and XSum test data. The best and second best performances over the basic transformer are in **bold** and underlined. ↑/↓ indicates the results are significantly higher/lower than Basic Transformer and ⇑/⇓ indicates the results are significantly higher/lower than Shuffling ($p < 0.01$ with bootstrap resampling test (Lin, 2004)).

Model	$\mathbf{D}_{early}$	$\mathbf{D}_{middle}$	$\mathbf{D}_{late}$
Lead-3	49.33	30.90	19.80
Oracle	49.51	47.02	43.81
Basic Transformer	44.30	31.91	22.65
– No Position Encoding	16.07	16.88	18.59
– Only Position Encoding	$48.65^{*\dagger\ddagger}$	30.97	19.70
Learned-Mixin	40.45	31.82	22.70
Shuffling	42.69	31.91	$22.99^{*\dagger\ddagger}$
Our Method	42.67	$32.18^{*\dagger\ddagger}$	$22.85^{*\dagger}$

Table 2: Avg. of ROUGE-1/2/L F1 scores on D_{early}, D_{middle} and D_{late}. Results significantly better than Basic Transformer on ROUGE-1/2/L are marked with $*$, †, and ‡ respectively.

is a method proposed lately for summarization lead bias demoting, and ***Learned-Mixin*** (Clark et al., 2019), which is a general debiasing method proposed to deal with NLP tasks when the type of data bias in the training set is known and bias-only model is available. In our case, the data bias is lead bias and the bias-only model is the transformer trained with only positional encoding as input.

Implementation Details: All the transformer-based models have the same setting as the standard transformer (Vaswani et al., 2017), with 6 layers, 8 heads per layer, and $d_{model} = 512$. We use Adam to train all the models with scheduled learning rate with warm-up (initial learning rate $lr = 2e - 3$). We choose the top-3 sentences to form the final summary for CNN/DM and the top-1 sentence for XSum due to the different average summary lengths. The class number of sentence position M is set to 10 and trade-off parameter β is set to 0.9 (searched from 0 to 1, by increasing 0.1 for each step). We tune these hyper-parameters on a "balanced" validation set sampled from the standard CNN/DM validation data.

3.3 Results and Analysis

Table 1 reports the performance of the chosen baselines and our proposal on CNN/DM test set, which has the same data distribution as the training data, and XSum test set, which is from another news resource and with much less lead bias than CNN/DM.

From the middle section of Table 1, we observe that if we withhold the position cues (*No Position Encoding*) by using only semantic representation as input, the model's performance drops considerably on CNN/DM, but remarkably increase on XSum. In contrast, if we merely use position cues as input (*Only Position Encodings*), the decrease of the performance on CNN/DM becomes much more modest, while there is substantial performance drop on XSum. These results confirm that positional signal is a rather important feature for bias-relied neural summarizers. However, relying too much on it will also limit model's generality when applied to the dataset with less bias than the training samples. Therefore, seeking strategies to balance the semantics and position features is crucial for the neural extractive summarization for news.

When we compare the lead bias demoting methods presented at the bottom of Table 1, our proposal and *Shuffling* give significant performance boosting on XSum, while *Learned-Mixin* results in performance decrease on both datasets. Comparing our method and *Shuffling* directly, while they are essentially equivalent on maintaining the performance on the in-distribution CNN/DM data (0.09 difference in terms of the average of ROUGE scores (ROUGE-Mean)), our method provides a significant improvement on XSum, and outperforms *Shuffling* and the basic transformer by 0.14 and 0.29 on ROUGE-Mean respectively. It is noteworthy that the transformer without position encoding achieves

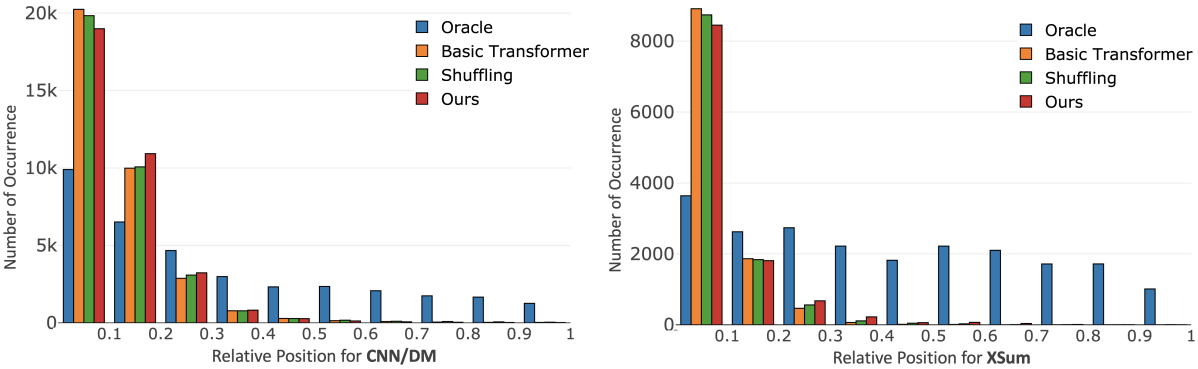


Figure 2: Relative position distributions of selected sentences in the original document of two testing corpora (CNN/DM and XSum), obtained by different lead bias demoting strategies.

the best performance on XSum. However, it is the worst system on in-distribution data. Throughout all the comparisons, our proposal can best balance the sentence position and content semantics.

To more deeply investigate the behavior of our demoting method on the documents whose summary sentences are from different document portions, we follow Grenander et al. (2019) to create three subsets, D_{early}, D_{middle}, D_{late}, from the CNN/DM testing set. Documents are ranked by the mean of their summary sentences' indices in ascending order, and then the top-ranked 100 documents, the 100 documents closest to the median, and the bottom-ranked 100 documents are selected to form D_{early}, D_{middle}, D_{late}[2]. Results in Table 2 show that even if our model does not match the basic transformer on documents in D_{early}, it does yield benefits for both D_{middle} and D_{late} with significant improvements, while the competitive baseline *Shuffling* only achieves that on D_{late}.

Position of Selected Content: To more explicitly investigate how well the prediction of different models fits the ground-truth sentence selection (*Oracle*), we compare the relative position of the selected content of our method with the undebiased model (*Basic Transformer*) and the most competitive debiased model (with *Shuffling*), as illustrated in Figure 2. We can observe that: (1) CNN/DM contains much more lead bias than XSum, shown by a more right-skewed histogram for *Oracle*. Thus, the basic transformer trained on it is also heavily impacted by the lead bias and tends to select sentences $\in [0, 0.1]$ with much higher probability even on the less biased XSum. (2) While *Shuffling* and our method can both effectively demote the extreme trend towards selecting sentences in the lead position, our method seems to be sightly better at encouraging the model to select sentences with higher relative position.

[2]Due to the common generation mechanism of oracles, the number of sentences in the oracle is not fixed. For fair comparison, we only consider adding documents with the oracle having exactly 3 sentences into D_{early}, D_{middle}, D_{late}.

4 Conclusion and Future Work

We propose a lead bias demoting method to make news extractive summarizers more robust across datastets, by optimizing a position prediction and a summarization component in an alternating way. Experiments indicate that our method improves model's generality on out-of-distribution data, while still largely maintaining its performance on in-distribution data. As such, it represents the best viable solution when at inference time input documents may come from an unknown mixture of datasets with different degrees of position bias.

For the future, we plan to explore more sophisticated and effective methods (e.g., adjusting the lead bias online) and infuse them together with neural abstractive summarization models, known to generate more succinct and natural summaries. Another interesting direction for future work can be exploring the potential of applying our proposed bias demoting strategy to other tasks, which can also be framed as the sequence labeling problem and possibly troubled by biases in the training data (e.g., Topic Segmentation (Xing et al., 2020) and Semantic Role Labeling (Ouchi et al., 2018)).

Acknowledgments

We thank the anonymous reviewers and the UBC-NLP group for their insightful comments and suggestions. This research was supported by the Language & Speech Innovation Lab of Cloud BU, Huawei Technologies Co., Ltd.

References

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Grenander, Yue Dong, Jackie Chi Kit Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6019–6024, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, pages 1693–1701. Curran Associates, Inc.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, Gothenburg, Sweden. Association for Computational Linguistics.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. 2019. Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3324–3335, Hong Kong, China. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Sachin Kumar, Shuly Wintner, Noah A. Smith, and Yulia Tsvetkov. 2019. Topics to avoid: Demoting latent confounds in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4153–4163, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. AAAI'17, page 3075–3081. AAAI Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Ani Nenkova, Sameer Maskey, and Yang Liu. 2011. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 3, Portland, Oregon. Association for Computational Linguistics.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019. Exploring domain shift in extractive text summarization. *CoRR*, abs/1908.11664.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528, Suzhou, China. Association for Computational Linguistics.

Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. Do we really need that many parameters in transformer for extractive summarization? discourse can help ! In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134, Online. Association for Computational Linguistics.

Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. 2020. Improving context modeling in neural topic segmentation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 626–636, Suzhou, China. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019a. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Danqing Wang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2019b. A closer look at data bias in neural extractive summarization models. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 80–89, Hong Kong, China. Association for Computational Linguistics.

DuReader$_{\text{robust}}$: A Chinese Dataset Towards Evaluating Robustness and Generalization of Machine Reading Comprehension in Real-World Applications

Hongxuan Tang[1]* **Hongyu Li**[2] **Jing Liu**[2]† **Yu Hong**[1]† **Hua Wu**[2] **Haifeng Wang**[2]
[1]School of Computer Science and Technology, Soochow University, China
[2]Baidu Inc., Beijing, China
{hxtang01, tianxianer}@gmail.com
{liujing46, lihongyu04, wu_hua, wanghaifeng}@Baidu.com

Abstract

Machine reading comprehension (MRC) is a crucial task in natural language processing and has achieved remarkable advancements. However, most of the neural MRC models are still far from robust and fail to generalize well in real-world applications. In order to comprehensively verify the robustness and generalization of MRC models, we introduce a real-world Chinese dataset – DuReader$_{\text{robust}}$. It is designed to evaluate the MRC models from three aspects: over-sensitivity, over-stability and generalization. Comparing to previous work, the instances in DuReader$_{\text{robust}}$ are natural texts, rather than the altered unnatural texts. It presents the challenges when applying MRC models to real-world applications. The experimental results show that MRC models do not perform well on the challenge test set. Moreover, we analyze the behavior of existing models on the challenge test set, which may provide suggestions for future model development. The dataset and codes are publicly available at https://github.com/baidu/DuReader.

1 Introduction

Machine reading comprehension (MRC) requires machines to comprehend text and answer questions about it. With the development of deep learning, the recent studies of MRC have achieved remarkable advancements (Seo et al., 2017; Wang and Jiang, 2017; Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020). However, previous studies show that most of the neural models are not robust enough (Jia and Liang, 2017; Ribeiro et al., 2018b; Talmor and Berant, 2019a; Welbl et al., 2020) and fail to generalize well (Talmor and Berant, 2019b).

To further promote the studies of robust and well generalized MRC, we construct a Chinese dataset – DuReader$_{\text{robust}}$ which comprises natural questions and documents. In this paper, we focus on evaluating the robustness and generalization from the following aspects, where robustness consists of over-sensitivity and over-stability:

(1) **Over-sensitivity** denotes that MRC models provide different answers to the paraphrased questions. It means that the models are overly sensitive to the difference between the original question and its paraphrased question. We provide an example in Table 1a.

(2) **Over-stability** means that the models might fail into a trap span that has many words in common with the question, and extract an incorrect answer from the trap span. Because the models overly rely on spurious lexical patterns without language understanding. We provide an example in Table 1b.

(3) **Generalization**. The well-generalized MRC models have good performance on both in-domain and out-of-domain data. Otherwise, they are less generalized. We provide an example in Table 1c.

In previous work, the above issues have been studied separately. In this paper, we aim to create a dataset namely DuReader$_{\text{robust}}$ to comprehensively evaluate the three issues of neural MRC models. Previous work mainly studies these issues by altering the questions or the documents. Ribeiro et al. (2018b); Iyyer et al. (2018); Gan and Ng (2019) evaluate the over-sensitivity issue via paraphrase questions generated by rules or generative models. Jia and Liang (2017); Ribeiro et al. (2018a); Feng et al. (2018); Talmor and Berant (2019a) focus on evaluating the over-stability issue by adding distracting sentences to the documents or reducing question word sequences. However, the altered questions and documents are not natural texts and rarely appear in the real-world applications. It is not clear that how the evaluation based

* This work was done while the first author was doing internship at Baidu Inc.

† Corresponding authors

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 955–963
August 1–6, 2021. ©2021 Association for Computational Linguistics

Passage 近年来，随着琥珀蜜蜡市场的兴起，蜜蜡与琥珀的价格都有不断上涨的趋势，其中蜜蜡首饰的价格一般是琥珀首饰价格的2–4倍，最近几年二者价格差距更大.....	**Passage** *In recent years, with the rise of the amber market, the price of amber keeps going up. The price of opaque amber is generally 2–4 times the price of clear amber ...*
Original Question 琥珀和蜜蜡哪一个比较贵 **Golden Answer**： 蜜蜡 **Predicted Answer**： 蜜蜡 ($BERT_{base}$)	**Original Question** *Which is more expensive, clear amber or opaque amber?* **Golden Answer** : *opaque amber* **Predicted Answer** : *opaque amber* ($BERT_{base}$)
Paraphrase Question 蜜蜡和琥珀哪个价格高 **Golden Answer**： 蜜蜡 **Predicted Answer**： 琥珀 ($BERT_{base}$)	**Paraphrase Question** *Which has the higher price, opaque amber or clear amber?* **Golden Answer** : *opaque amber* **Predicted Answer** : *clear amber* ($BERT_{base}$)

(a) An example illustrates the over-sensitivity issue, where $BERT_{base}$ gives different predictions to the original question and the paraphrased question.

Passage 包粽子的线以前人们认为是来自麻叶子，其实是棕榈树，粽子的音就来自棕叶子。	**Passage** *Many people argue that the zongzi (rice dumpling) leaves are made of hemp. Actually, it is the palm tree, the real origin, that endows zongzi with the special pronunciation.*
Question 包粽子的线来自什么 **Golden Answer**： 棕榈树 **Predicted Answer**： 麻叶子 ($BERT_{base}$)	**Question** *What is the raw material of zongzi leaves?* **Golden Answer** : *palm tree* **predicted Answer** : *hemp* ($BERT_{base}$)

(b) An example illustrates the over-stability issue. The underlined span in the passage appears as a trap because it has many words in common with the question. $BERT_{base}$ falls into the trap.

Passage *cos(2x)'=-sin(2x)*(2x)'=-2sin(2x)* 属于复合函数的求导。	**Passage** *cos(2x)'=-sin(2x)*(2x)'=-2sin(2x) This is the derivative of a compound function.*
Question *cos2x*的导数是多少？ **Golden Answer** : *-2sin(2x)* **Predicted Answer** : *-sin(2x)* ($BERT_{base}$)	**Question** *What is the derivative of cos2x?* **Golden Answer** : *-2sin(2x)* **Predicted Answer** : *-sin(2x)* ($BERT_{base}$)

(c) An example illustrates the generalization issue. Although $BERT_{base}$ is sufficiently trained on large-scale open-domain data, it fails to predict the answer to a math question.

Table 1: The examples of over-sensitivity, over-stability and generalization issues.

on such unnatural texts can help the improvements of the neural models in real-world applications. By contrast, all the instances in $\text{DuReader}_{\textbf{robust}}$ are natural texts and collected from the Baidu search.

We conduct extensive experiments based on $\text{DuReader}_{\textbf{robust}}$. The experimental results show that the models based on pre-trained language models (LMs) (Devlin et al., 2019; Sun et al., 2019; Liu et al., 2019) do not perform well on the challenge set. Besides, we have the following findings on the behaviors of the models: (1) if a paraphrased question contains more words rephrased from the original question, it is more likely that MRC models provide different answers; (2) the trap spans which share more words with the questions easily mislead MRC models; (3) domain knowledge is a key factor that affects the generalization ability of MRC models.

2 Dataset: $\text{DuReader}_{\textbf{robust}}$

$\text{DuReader}_{\textbf{robust}}$ is built on DuReader, a large-scale Chinese MRC dataset (He et al., 2018). In DuReader, all questions are issued by real users of Baidu search, and the document-level contexts are collected from search results. In $\text{DuReader}_{\textbf{robust}}$, we select entity questions and paragraph-level contexts from DuReader. We further employ crowdworkers to annotate the answer span conditioned on the question and the paragraph-level context [1]. Additionally, we used a mechanism to ensure data quality, where 10% of the annotated data will be randomly selected and reviewed by linguistic experts. If the accuracy is lower than 95%, the crowdworkers need to revise all the answers until the accuracy for the randomly selected data is higher than 95%.

Dataset	len(p)	len(q)	len(a)	#
Train	291.88	9.19	5.39	14,520
Development	288.16	9.38	6.66	1,417
Test	285.36	9.41	6.55	1,285
Challenge	132.09	11.97	7.33	3,556
All				**20,778**

Table 2: The statistics of $\text{DuReader}_{\textbf{robust}}$.

[1] The instances which have insufficient contexts for answering the questions are discarded.

Answer Type	%	Examples
Date	24.7	15分钟 (15 minutes)
Number	17.5	53.28厘米 (53.28cm)
Interval	11.8	1%至5% (1% to 5%)
Person	8.8	成龙 (Jackie Chan)
Organization	7.5	湖南卫视 (Hunan Satellite TV)
Money	7.0	2.7亿美元 (270 million dollars)
Location	6.0	北京 (Beijing)
Software	2.2	百度地图 (Baidu Map)
Item	1.6	华为P9 (Huawei P9)
Other	12.9	管理学 (Management Science)

Table 3: The frequency distribution and examples of different answer types in DuReader$_{\textbf{robust}}$.

Eventually, we collect about 21K instances for DuReader$_{\textbf{robust}}$, each of which is a tuple $\langle q, p, A\rangle$, where q is a question, p is a paragraph-level context containing reference answers A. Similar to the existing MRC datasets, DuReader$_{\textbf{robust}}$ consists of training set, in-domain development set and in-domain test set, whose sizes are 15K, 1.4K and 1.3K respectively. Besides, DuReader$_{\textbf{robust}}$ contains a challenge test set, in which 3.5K instances are created to evaluate the robustness and generalization of MRC models. The challenge test set can be divided into three subsets including over-sensitivity set, over-stability set and generalization set. Table 2 shows the statistics of DuReader$_{\textbf{robust}}$. Besides, DuReader$_{\textbf{robust}}$ covers a wide range of answer types (e.g. date, numbers, person, etc.). The frequency distribution and examples of the answer types are shown in Table 3. Next, we will present our way to construct the three subsets in the challenge test set.

2.1 Over-sensitivity Subset

We build the over-sensitivity subset in the following way. First, we sample a subset of instances $\{\langle q, p, A\rangle\}$ from the in-domain test set of DuReader$_{\textbf{robust}}$. For each question q, we obtain its N paraphrases $\{q'_1, q'_2, ..., q'_N\}$ using the paraphrase retrieval toolkit (See Appendix A for further details). To ensure the paraphrase quality, we employ crowd-workers to discard all false paraphrases. Then, we replace q with the paraphrased question q'_i, and keep the original context p and answers A unchanged. This leads to the new instances $\{\langle q'_i, p, A\rangle\}$, and they are used as the model-independent instances in the over-sensitivity subset. Besides, we also employ a model-dependent way to collect instances. Specifically, we use paraphrased instances to attack the MRC models based on ERNIE (Sun et al., 2019) and RoBERTa (Liu et al., 2019). If one of the models gives a different prediction from the predicted answer of the original question, we adopt the instance, otherwise we discard it. The instances collected in the above model-dependent and model-independent ways constitute the over-sensitivity subset. The over-sensitivity subset consists of 1.2K instances. The number of model-independent instances is equal to that of model-dependent instances. Table 1a shows an example in the over-sensitivity subset.

Algorithm 1: Annotate an instance for over-stability subset

Input: $\{\langle q, p, A\rangle\}$ tuple
Output: $\{\langle q', p, A'\rangle\}$ tuple or null
Identify the named entities $\{e_1, ..., e_n\}$ along with their entity types in p
Keep the named entities $\{e_i, ..., e_m\}$ with the same types as A
if $1 < m < k$ **then**
  if *linguistic experts consider the passage* p *contains a trap* **then**
    annotate a new question q' and answers A'
    A and A' share the same named entity type
    return $\{\langle q', p, A'\rangle\}$
  else return null;
else return null;

2.2 Over-stability Subset

Intuitively, a trap span that has many words in common with the questions may easily mislead MRC models. Following this intuition, the over-stability subset is constructed as follows. First, we randomly select a set of instances $\langle q, p, A\rangle$ from DuReader. In general, a trap span may contain non-answer named entities of the same type as the reference answers A. This is because over-stable models usually rely on spurious patterns that match the correct answer types. Thus, we use a named entity recognizer [2] to identify all named entities in p along with their entity types. We keep the corresponding instance, if there are non-answer named entities that are of the same type as A. Then, we ask linguistic experts to annotate a new question q' and answers A', if they consider p contains trap spans. A and A' share the same named entity type. The annotated question q' has a high level of lexical overlap with a trap span that does not contain A. We say $\{\langle q', p, A'\rangle\}$ can be considered as a candidate instance. Each candidate instance is used to attack one of the MRC models based on ERNIE (Sun et al., 2019) and RoBERTa (Liu et al.,

[2] https://ai.baidu.com/tech/nlp_basic/lexical

	In-domain dev set		In-domain test set		Challenge test set	
	EM	F1	EM	F1	EM	F1
BERT$_{\text{base}}$	71.20	82.87	67.70	80.85	37.57	53.86
ERNIE 1.0$_{\text{base}}$	68.73	81.12	66.72	80.50	36.75	55.64
RoBERTa$_{\text{large}}$	74.17	86.02	71.20	84.16	45.02	62.83
Human			78.00	89.75	72.00	86.43

Table 4: Comparing MRC baselines to human on the development, test and all challenge sets.

	Over-Sensitivity		Over-Stability		Genera-lization	
	EM	F1	EM	F1	EM	F1
BERT$_{\text{base}}$	53.31	69.30	16.78	38.40	36.41	50.15
ERNIE 1.0$_{\text{base}}$	58.10	73.89	17.27	38.34	32.86	52.84
RoBERTa$_{\text{large}}$	55.24	75.16	28.18	47.03	46.03	61.67

Table 5: The results on the three subsets of the challenge set.

2019). The candidate instance will be used to construct an overstability subset, if one of the model fails. Algorithm 1 shows the detailed procedure (See Appendix B for details). As a result, we have 0.8K instances to evaluate over-stability. Table 1b shows an example from the over-stability subset.

2.3 Generalization Subset

The in-domain test set consists merely of in-domain data (i.e., the distribution is the same as the one in the training and development sets). In order to evaluate the generalization ability of MRC models, we construct a generalization subset which comprises out-of-domain data. The out-of-domain data is collected from two vertical domains. The details are as follows.

Education We collect educational questions and documents from Baidu search, and we ask crowdworkers to annotate 1.2K high-quality tuples $\langle q, p, A \rangle$. The topics include mathematics, physics, chemistry, language and literature. Table 1c shows an example.

Finance Following Fisch et al. (2019), we leverage a dataset that was originally designed for information extraction in the finance domain for MRC. We obtain 0.4K instances of financial reports this way. The construction details are presented in Appendix C.

3 Experiments

3.1 Baselines and Evaluation Metrics

We consider three baseline models in the experiments. They are based on different pre-trained language models, including BERT$_{\text{base}}$ (Devlin et al., 2019), ERNIE 1.0$_{\text{base}}$ (Sun et al., 2019) and RoBERTa$_{\text{large}}$ (Liu et al., 2019). In Appendix D, we set the hyperparameters of our baseline models.

Following Rajpurkar et al. (2016), we use exact match (EM) and F1-score to evaluate the held-out accuracy of an MRC model. All the metrics are calculated at Chinese character level, and we normalize both the predicted and true answers by removing spaces and punctuation marks.

	DPR (%)
BERT$_{\text{base}}$	22.73
ERNIE 1.0$_{\text{base}}$	19.88
RoBERTa$_{\text{large}}$	16.44

Table 6: The DPRs of baselines on the over-sensitivity subset.

3.2 Main Results

Table 4 shows the baseline results on the in-domain development set, in-domain test set, and challenge test set. The baseline performance is close to human performance on the in-domain test set, whereas the gap between baseline performance and human performance on the challenge test set is much larger. In Appendix E, we describe the method for calculating human performance.

We further evaluate the baselines on the three challenge subsets for over-sensitivity, over-stability and generalization separately. Table 5 shows the results. We have found that baseline performance declines significantly for over-stability and generalization subsets (compared to the "In-domain test set" in Table 4). In contrast, the baseline performance degrades less significantly on the over-sensitivity subset, although there is still a noticeable gap.

3.3 Discussion 1: Over-sensitivity

First, we calculate the different prediction ratios (DPRs) of the baselines on the over-sensitivity subset. DPR measures the percentage of the paraphrased questions that yield different predictions. DPR is formulated in Appendix F . Table 6 presents the DPRs of the baselines on the over-sensitivity subset. The baselines obtained around 16% to 22% DPRs, which demonstrates that the baselines are sensitive to part of the paraphrased questions.

Second, we examine a hypothesis - if a paraphrased question contains more words rephrased from the original question, the MRC model is more likely to produce different answers. To measure how similar paraphrased questions are to the original questions, we use the F1-score. A low F1-score

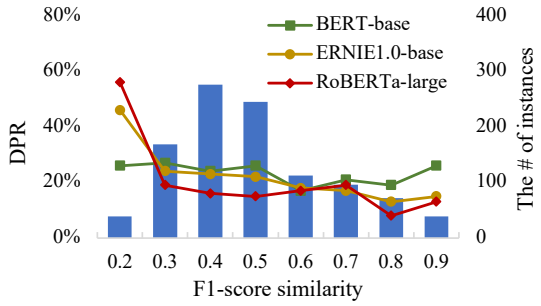


Figure 1: The correlation between DPR and F1-score based question similarity on the over-sensitivity subset.

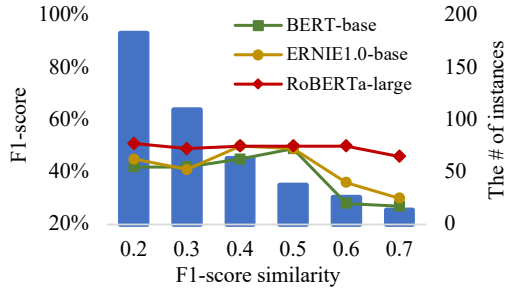


Figure 2: The correlation between the model performance and question-trap similarity on the over-stability subset.

means that many words in the original question have been rephrased. We divide the paraphrased questions into buckets based on how similar they are to the original questions, and we then examine whether there is correlation between DPR and F1-score similarity. Based on Figure 1, we can observe that the DPRs of all the baselines are negatively correlated with the F1-score similarity between the original and paraphrased questions. The results confirm the hypothesis.

3.4 Discussion 2: Over-stability

MRC models might be easily misled by trap spans that share many words with the questions. We examine whether there is a correlation between MRC performance (F1-score) and question-trap similarity in this section. Based on the similarity between trap spans and questions, we divide trap spans into buckets. According to Figure 2, the performance of the base models decreases as similarity increases and the large model (RoBERTa$_{\text{large}}$) is less over-stable than the base ones.

3.5 Discussion 3: Generalization

Table 7 shows the baseline performance in the the domains of finance and education. We can observe

	Finance		Education	
	EM	**F1**	**EM**	**F1**
BERT$_{\text{base}}$	30.73	51.16	38.70	50.83
ERNIE 1.0$_{\text{base}}$	26.53	50.53	34.67	53.11
RoBERTa$_{\text{large}}$	40.22	61.16	47.77	61.82

Table 7: The performance of baselines in the domains of education and finance.

Topcis	**EM**	**F1**	**#**
Math	19.85	34.63	136
Chemistry	37.46	53.88	323
Language	44.31	61.18	255
Others	69.63	79.28	438
All	49.13	62.88	1152

Table 8: The performance of baselines on different topics in the domain of education.

that the baselines perform poorly for both domains. Additionally, we examine how baseline models behave in the education domain. Table 8 shows the performance of RoBERTa$_{\text{large}}$ on the four topics in the education domain. The model performs much worse when it comes to math and chemistry, since these topics are rare in the training set. The results of this analysis suggest that domain knowledge is a key factor affecting the generalization ability of MRC models. More discussion can be found in Appendix G.

4 Conclusion

In this paper, we create a Chinese dataset – DuReader$_{\textbf{robust}}$ and use it to evaluate both the robustness and generalization of the MRC models. Its questions and documents are natural texts from Baidu search. This presents the robustness and generalization challenges in the real-world applications. Our experiments show that the MRC models based on the pre-trained LMs do not perform well on DuReader$_{\textbf{robust}}$ challenge set. We also conduct extensive experiments to examine the behaviors of the MRC models on the dataset and provide insights for future model development.

Acknowledgement

This research work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900) and National Natural Science Foundation of China (Grants No.61773276 and No.62076174). The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Jing Liu and Yu Hong are the corresponding authors of the paper.

Ethical Considerations

We aim to provide researchers and developers with a dataset $\text{DuReader}_{\text{robust}}$ to improve the robustness and generalization ability of MRC models. We also take the potential ethical issues into account. (1) All the instances in the $\text{DuReader}_{\text{robust}}$ have been desensitised. (2) Regarding to the issue of labor compensation, we make sure that all the crowdsourcing workers are fairly compensated.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018a. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018b. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 856–865. Association for Computational Linguistics.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Alon Talmor and Jonathan Berant. 2019a. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4911–4921. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019b. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4911–4921. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. Machine comprehension using match-lstm and answer pointer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Johannes Welbl, Pasquale Minervini, Max Bartolo, Pontus Stenetorp, and Sebastian Riedel. 2020. Undersensitivity in neural reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1152–1165. Association for Computational Linguistics.

A Paraphrase Retrieval Toolkit

We use a paraphrase retrieval toolkit to obtain paraphrase questions. The toolkit is used internally at Baidu, and our manual evaluations show that the accuracy of the retrieval results is around 98%. The paraphrase retrieval toolkit consists of two basic modules as follows.

- **Paraphrase Candidate Retriever** The retriever is a light-weight module. Given a question, the retriever will retrieve top-k paraphrase candidates from the search logs of Baidu Search.
- **Paraphrase Candidate Re-ranker** The re-ranker is a model fine-tuned from ERNIE (Sun et al., 2019) by using a set of manually labeled paraphrase questions. Given a set of retrieved paraphrase candidates, the re-ranker will estimate the semantic similarity between the original question and the paraphrase candidates. If the semantic similarity is higher than a pre-defined threshold, the candidate will be used as a paraphrased question.

B The Illustration of Annotating Over-stability Instances

Figure 3 illustrates the annotation of an over-stability instance. In the instance, the answer to the original question is 30-40 minutes. The entity type of 5-10 minutes is the same as 30-40 minutes. The annotator raise a new question by revising the original question, the answer to the new question is 5-10 minutes. The sentence contains 30-40 minutes has many words in common with the new question, and it is considered as a trap sentence. The new question may mislead the model to predict the answer to the new question as 30-40 minutes.

C The Construction of Finance Data

We leverage a dataset that is originally designed for information extraction in finance domain. The original dataset contains the full texts of the financial reports as documents and the structured data that is extracted from the texts. Then, we use templates to generate questions for each data field in the structured data. Finally, we use these constructed instances for MRC. Each instance contains (1) a question generated from a template for a data field, (2) an answer that is the value in the data field and (3) a document from which the value (i.e. answer) is extracted.

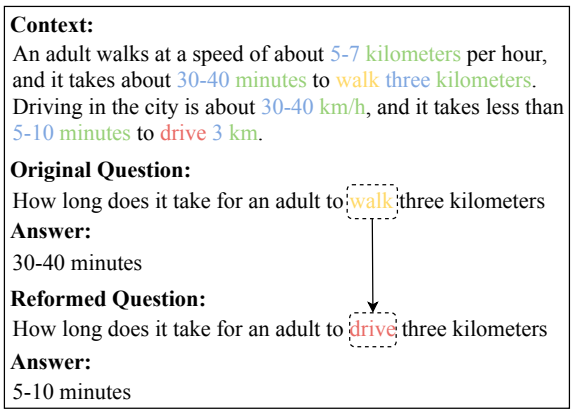


Figure 3: The illustration of annotating an over-stability instance.

D Hyperparameters

We use a number of pre-trained language models in our baseline systems. When fine-tuning different pre-trained language models, we use the same hyperparameters. The settings of hyperparameters are as follows. The learning rate is set to 3e-5 and the batch size is 32. We set the number of epochs to 5. The maximal answer length and document length are set to 20 and 512, respectively. We set the length of document stride to 128. All experiments are conducted on 4 Tesla P40 GPUs.

E Human Performance

We evaluate human performance on both the in-domain test set and challenge test set. We randomly sample two hundred instances from the in-domain test set, and three hundred instances from the challenge test set. We ask crowdworkers to provide answers to the questions in the sampled instances. Then, we use EM and F1-scores of these annotated answers as human performance.

F Different Prediction Rate

Different prediction rate (DPR) measures the percentage of paraphrase questions whose predictions are different from the original questions. Formally, we define DPR of a neural model $f(\theta)$ on a dataset D as follows.

$$DPR_D(f(\theta)) = \frac{\sum_{(q,q\prime)\in Q} \mathbb{1}[f(\theta;q) \neq f(\theta;q\prime)]}{\|Q\|}$$

where, $f(\theta;q)$ denotes the prediction of the trained MRC model $f(\theta)$. Q represents a set of pairs of

Types	# of changes (%)	# of same (%)
WR	1 (12.50)	7 (87.50)
RF	0 (00.00)	4 (100.00)
SS	6 (17.14)	29 (82.85)
AD	7 (23.33)	23 (76.66)
CO	7 (30.43)	16 (69.56)

Table 9: Distributions of paraphrases and DPRs.

Question Types	EM	F1	#
Company abbreviations	0	31.15	18
Pledgee	80.76	89.96	26
Pledgor	0	24.62	25
The pledge amount	18.36	53.84	98
Others (e.g. pledge date)	47.91	58.97	48
All	28.83	54.05	215

Table 10: The performance of RoBERTa$_{\text{large}}$ on the five topics in the domain of financial reports.

original question q and paraphrased question $q\prime$ in dataset D, and $\mathbb{1}[*]$ is an indicator function. A high DPR score means that the MRC model is overly sensitive to the paraphrased question $q\prime$, otherwise insensitive.

G Experimental Analysis

G.1 Over-sensitivity Analysis

We further analyze the prediction results to figure out what kind of paraphrases lead to different predictions. Five types of paraphrasing phenomena have been found, including (1) word reordering (WR), (2) replacement of function words (RF), (3) substitution by synonyms (SS), (4) inserting or removing content words (AD), and (5) more than one previously defined types happen in one paraphrase (CO). We randomly sample one hundred instances from the over-sensitivity subset and analyze the changes of the predictions by ERNIE (Sun et al., 2019). As shown in Table 9, most of changed predictions come from AD and CO. This analysis suggests that the models are sensitive to the changes of content words.

G.2 Generalization Analysis

In previous section, we have already analyzed the behaviors of baseline systems on education domain. In this section, we conduct analysis on financial domain. The data of financial domain contains management changes and equity pledge. The performance of RoBERTa$_{\text{large}}$ on management changes and equity pledge is 68.63% and 49.15% respectively. The model generalizes well on management changes, since the training set contains the relevant knowledge about asking person names. By contrast, the model performs worse on equity pledge. We classify the instances of equity pledge into five sets according to the question types. Table 10 shows the performance of RoBERTa$_{\text{large}}$ on the five question types. We can observe that the model performs the worst on the questions about company abbreviations, pledgee and pledgor, since there is little domain knowledge in the training set. By contrast, the model performs better on the questions about amount and date, since the model has already learnt relevant knowledge in the training set.

Sequence to General Tree: Knowledge-Guided Geometry Word Problem Solving

Shih-hung Tsai, Chao-Chun Liang, Hsin-Min Wang, Keh-Yih Su
Institute of Information Science, Academia Sinica, Taiwan
{doublebite,ccliang,whm,kysu}@iis.sinica.edu.tw

Abstract

With the recent advancements in deep learning, neural solvers have gained promising results in solving math word problems. However, these SOTA solvers only generate binary expression trees that contain basic arithmetic operators and do not explicitly use the math formulas. As a result, the expression trees they produce are lengthy and uninterpretable because they need to use multiple operators and constants to represent one single formula. In this paper, we propose sequence-to-general tree (S2G) that learns to generate interpretable and executable *operation trees* where the nodes can be formulas with an arbitrary number of arguments. With nodes now allowed to be formulas, S2G can learn to incorporate mathematical domain knowledge into problem-solving, making the results more interpretable. Experiments show that S2G can achieve a better performance against strong baselines on problems that require domain knowledge.[1]

1 Introduction

Math word problem (MWP) solving is a special subfield of question answering. It requires machine solvers to read the problem text, understand it, and then compose the numbers and operators into a meaningful equation (as shown in Table 1). This process, even for the simplest problem in elementary school, demands language understanding and numerical reasoning capabilities, making this task a long-standing challenge in AI (Bobrow, 1964; Zhang et al., 2019).

As with any QA task, solving an MWP requires the introduction of external knowledge or domain knowledge (Mishra et al., 2020). However, current state-of-the-art solvers (Xie and Sun, 2019; Zhang et al., 2020; Wu et al., 2020) do not address this issue explicitly. They learn to map the problem text into binary expression trees regardless of whether it requires any knowledge. This is counterintuitive for problems that need math concepts or formulas. As illustrated in Figure 1(a), without explicitly using the corresponding area formula, the expression tree for the problem is lengthy and uninterpretable.

To address this issue, we propose a sequence-to-general tree (S2G) architecture where the nodes

Problem: The outer radius and the inner radius of a circular annulus are 5m and 3m repsectively. Find the area of this circular annulus.
Equation: $x = 5 * 5 * 3.14 - 3 * 3 * 3.14$ **Answer:** 50.24
With formula: x = circle_area(5) - circle_area(3)

Table 1: Example problem that requires geometry knowledge.

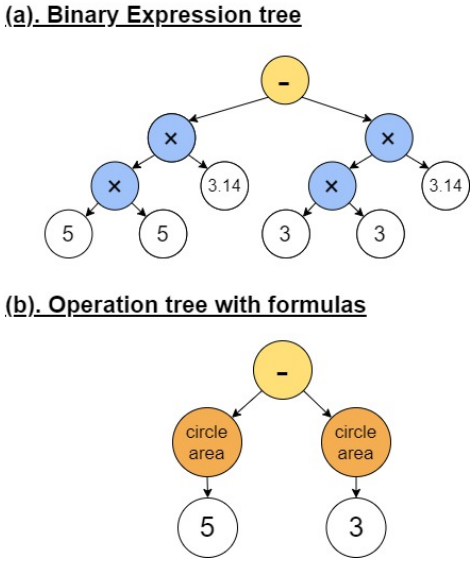


Figure 1: (a). binary expression tree and (b). operation tree along with formulas for the problem in Table 1.

[1]Data and code are available at the GitHub repository: https://github.com/doublebite/Sequence-to-General-tree/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 964–972
August 1–6, 2021. ©2021 Association for Computational Linguistics

can be arbitrary math concepts or formulas with arbitrary number of arguments. In this way, our S2G model can learn to map the problem text into executable operation trees that contain different formulas across different domains. For example, S2G can learn to generate tree nodes that contain the required geometry formula for circles, as shown in Figure 1(b), making the result more intuitive and explainable.

In addition, we propose a knowledge-guided mechanism to guide tree-decoding using a mathematical knowledge graph (KG). To evaluate our model, we also construct a middle-sized dataset consisting of 1,398 geometry word problems which require a diversified set of formulas. Experimental results show that our S2G model can provide better performance and more interpretable results against strong baselines on problems that require domain knowledge.

The main contributions of this paper are:

1. We propose a seq-to-general tree model that learns to map the problem text into operation trees where the nodes can be formulas with arbitrary number of arguments. This helps to incorporate domain knowledge into problem solving and produce interpretable results.

2. We design a knowledge-guided mechanism that guides tree decoding using mathematical knowledge graphs and GNNs.

3. We curate a middle-sized dataset that contains 1,398 geometry word problems. In addition, we annotate them with detailed formulas that can be readily converted into operation trees.

2 Seq2seq v.s. Seq2tree v.s. Seq2general

Our goal is to design a sequence-to-general tree model that learns to map the problem text into its corresponding operation tree. Before diving into the model, we first compare the decoding mechanisms between seq-to-seq, seq-to-tree and our seq-to-general tree solvers. Figure 2 illustrates the tree decoding process of these three types of model, respectively.

For seq2seq models, their decoder basically does two things: (1) predicting the current output and (2) generating the next state. These two steps can be conditioned on different information including the current state, the current input, or a context vector calculated using attention. The decoder would repeat these two steps until it outputs an end token.

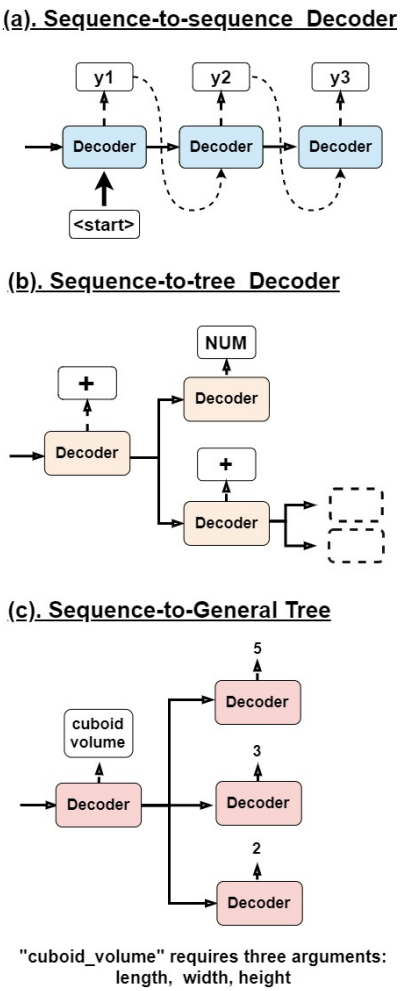


Figure 2: Comparison between three types of decoding: (a) seq2seq, (b) seq2tree, and (c) seq-to-general tree.

For seq2tree models, however, this process is slightly different. The decoder predicts the current output as in seq2seq, but it will decide whether to generate the next state based on the current output. If the current output is a arithmetic operator, the decoder knows it should produce two child states, and these states are used to expand into its left and right children. If the current output is a number, then the decoder would end the decoding process, so the current node becomes a leaf node. As a result, the whole decoding process resembles generating an expression tree in a top-down manner.

In our work, we generalize the decoding process by making the decoder produce a variable number of children based on the type of the current output. If the output is a number or operator, the decoder would produce zero or two child states as before. If the output is a formula, the decoder will generate the pre-specified number of child states for this formula.

3 Sequence-to-General Tree Model

In this section, we give a detailed description for each part of our S2G model.

3.1 Encoder

The main function of the encoder is to encode the problem text $P = (x_1, x_2, ..., x_n)$ into a sequence of hidden states $(h_1, h_2, ..., h_n)$ and their summary state $h_{encoder}$. The hidden states h_1 to h_n are expected to contain the information for each input token x_1 to x_n, while the summary state $h_{encoder}$ is expected to capture the overall information of the problem.

Specifically, we use bidirectional gated recurrent units (GRU) (Cho et al., 2014) as our encoder. Given the current input x_t, the previous state h_{t-1}, and the next state h_{t+1}, the current state $h_t \in (h_1, h_2, ..., h_n)$ can be calculated with:

$$\overrightarrow{h_t} = \mathrm{GRU}(x_t, \overrightarrow{h_{t-1}}), \quad (1)$$

$$\overleftarrow{h_t} = \mathrm{GRU}(x_t, \overleftarrow{h_{t+1}}), \quad (2)$$

where the arrows represent different directions in the bidirectional encoding. After calculating the hidden state for each input token, we combine the last state of the forward and backward directions to get the summary state for the encoder:

$$h_{encoder} = \overleftarrow{h_0} + \overrightarrow{h_n} \quad (3)$$

3.2 Geometry Knowledge Graph

To incorporate domain knowledge into problem solving, we propose to utilize the knowledge from mathematical knowledge graphs. The main idea is that given a formula predicted as the current node, we could use the physical meaning of its arguments to help us better predict its children. For example, if the current node is the formula for rectangle area, then we know its child nodes should be related to "length " and "width". We can thus use the node embeddings of "length" and "width" from a geometry KG to provide additional information for our solver.

We manually collect a geometry knowledge graph which contains the common geometry shapes (e.g., square, circle) and their geometry quantities (e.g., area, length), and we link these nodes to each other if they belong to the same shape. To embed this KG, we employ a graph convolutional network (GCN) (Kipf and Welling, 2017) that transforms the KG into some vector space and calculates the embedding of each node. Given the feature matrix X and the adjacency matrix A of the KG, we use a two-layer GCN to encode it as follows:

$$Z = \mathrm{GCN}(X, A), \quad (4)$$

where $Z = (z_1, ..., z_n)$ are the node embeddings for each node in the graph. Then, we can use the embedding to represent the physical meaning of a certain formula argument in the decoding process.

3.3 General Tree Decoder

In the decoding stage, the decoder learns to produce the target operation trees in a recursive manner. It first predicts the current output y_t in order to determine the number of children of the current node. Given the current decoder state s_t, the embedding of the last output $e_{(y_{t-1})}$, and the node embedding z_t which represents the physical meaning in the knowledge graph, the probability of the current output $P(y_t)$ is calculated using:

$$c_t = \mathrm{Attention}(\mathrm{e}_{(\mathrm{y_{t-1}})}, \mathrm{h}_1^{\mathrm{n}}) \quad (5)$$

$$z'_t = \mathrm{Attention}(\mathrm{z_t}, \mathrm{h}_1^{\mathrm{n}}) \quad (6)$$

$$P(y_t) = \mathrm{Softmax}(W_y[s_t; e_{(y_{t-1})}; c_t; z'_t]), \quad (7)$$

where h_1^n is the encoder states $(h_1, ..., h_n)$, c_t is the context vector of $e_{(y_{t-1})}$ with respect to h_1^n, and z'_t is another context vector calculated using the node embedding z_t and h_1^n. Specifically, we use additive attention (Bahdanau et al., 2015) to calculate these context vectors and use $h_{encoder}$ as the first decoder state s_0. Given the probability $P(y_t)$, we can then determine the output token $\hat{y}_t$:

$$\hat{y}_t = \mathrm{argmax} P(y_t). \quad (8)$$

Next, we predict the child states conditioned on the required number of children for $\hat{y}_t$. Unlike previous binary-tree decoders that use two distinct DNNs to predict the left and right children respectively (Xie and Sun, 2019; Zhang et al., 2020; Wu et al., 2020), we employ a GRU to predict a variable number of children. Given the current state s_t, its child states $s_{t_1}, ..., s_{t_n}$ are generated in a recurrent manner:

$$s_{t_i} = \mathrm{Decoder}(s_{t_{i-1}}; e_{(y_t)}; c_t), \quad (9)$$

where we generate the first child s_{t_1} using s_t, and the following child state s_{t_i} using its previous sibling $s_{t_{i-1}}$until we reach the required number of children. The decoder is basically a GRU followed

by a linear projection layer and an activation function:

$$s'_{t_i} = \text{GRU}([e_{(y_t)}; c_t], s_{t_{i-1}}), \quad (10)$$

$$s_{t_i} = \text{ReLU}(W_s s'_{t_i}), \quad (11)$$

where the input of GRU is the concatenation of $e_{(y_t)}$ and c_t, W_s is the linear projection layer, and ReLU is used as the activation function. After getting these child states, we push them into a stack and repeat the steps from Equation (5) to Equation (11) until all the states are realized into tokens.

3.4 Training Objective

For a problem and operation tree pair (*P, T*), we follow previous seq2tree work (Xie and Sun, 2019; Wu et al., 2020) and set our objective to minimize the negative log likelihood:

$$L(T, P) = \sum_{t=1}^{n} -logP(y_t|s_t, P, KG). \quad (12)$$

4 Dataset

To evaluate our S2G model on problems that require formulas, we curate a middle-sized dataset, *GeometryQA*, that contains 1,398 geometry word problems. These problems are collected from Math23K (Wang et al., 2017) using the keywords of common geometric objects (e.g., circle, square, etc.) and their shapes (e.g., rectangular, circular, etc.). Then, we re-annotate each problem with their associated formulas if the problem belongs to one of the six major shapes: *square*, *cubic*, *rectangle*, *cuboid*, *triangle* and *circle*. Table 2 shows the overall statistics of GeometryQA and Table 7 in Appendix B shows the 11 formulas we used to annotate these problems.

Note that not all problems in GeometryQA are annotated with formulas. About 16% of the problems belong to other shapes (e.g., *parallelogram*, *rhombus*, etc.) which currently are not covered in our formula set. About 40% are problems that contain geometric keywords but do not actually require any formulas. Table 3 shows such an example. We use these problems to test the robustness of our model. That is, S2G has to learn to apply the correct formulas or equations from misleading keywords (as shown in Table3) and has to learn to generate both binary expression trees and operation trees as a whole.

GeometryQA	
Number of problems	1,398
Number of sentences/words	5.4k / 41.1k
Vocabulary size	2,872
Annotated with formulas	604 (43.20%)
Problems of other shapes	225 (16.09%)
Formulas not required	569 (40.70%)

Table 2: Dataset statistics of GeometryQA

Problem: The perimeter of a rectangular swimming pool is 300 m. If you place a chair every 10 m all the way around its perimeter, how many chairs do you need?
Equation: $x = 300/10$ **Answer:** 30

Table 3: Example problem that contains misleading keywords (perimeter, rectangular) but do not require any geometry formulas.

5 Experiments

5.1 Implementation Details

We implement our S2G model and the GNN module using Pytorch[2] and Pytorch Geometric[3]. We set the dimension of word embedding to 128 and the dimension of the hidden state of GRU and GNN to 512. The dropout rate (Srivastava et al., 2014) is set to 0.5 and the batch size is 64. For optimization, we use ADAM (Kingma and Ba, 2015) with a learning rate of 10^{-3} and a weight decay of 10^{-5}. Besides, we use a learning rate scheduler to reduce the learning rate by half every 20 epochs. During evaluation, we use beam search (Wiseman and Rush, 2016) with a beam size of 5.

5.2 Experimental Results on GeometryQA

We evaluate our S2G model on GeometryQA to check whether it can learn to predict the corresponding operation tree for the geometry word problems. Table 4 shows the results of our S2G against other seq2tree SOTA models. S2G is trained using the re-annotated equations that contain formulas, while the baselines are trained using the original equations.

First, we find that S2G has about 3.8% perfor-

[2]https://pytorch.org/

[3]https://pytorch-geometric.readthedocs.io/en/latest/

mance gain over its baselines (with p-value < 0.01). We attribute this to the fact that operation trees are easier to learn and generate since they are less lengthy and complex than binary expression trees. Hence, there is a better chance for S2G to produce the correct trees and arrive at the correct answers.

Second, there is a small performance gain by adding Geometry KG. However, the improvement is not significant (with p-value$\approx$0.8). We guess that is because the dataset currently has only six geometric objects, which is not complex enough to show the effectiveness of adding knowledge graphs.

Model	Accuracy(%)
KA-S2T (Wu et al., 2020)	49.61%
GTS (Xie and Sun, 2019)	51.01%
S2G	**54.79%**
S2G + Geometry KG	**54.99%**

Table 4: Answer accuracy of S2G and other SOTA seq2tree models on GeometryQA (all evaluated with 5-fold cross validation).

6 Conclusion

In this work, we proposed a sequence-to-general tree model (S2G) that aims to generalize previous seq2tree architectures. Our S2G can learn to generate executable operation trees where the nodes can be formulas with arbitrary number of arguments. By explicitly generating formulas as nodes, we make the predicted results more interpretable. Besides, we also proposed a knowledge-guided mechanism to guide the tree decoding using KGs and constructed a dataset in which problems are annotated with associated formulas. Experimental results showed that our S2G model can achieve better performance against strong baselines.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Daniel G Bobrow. 1964. Natural language input for a computer problem solving system.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Edward A Feigenbaum, Julian Feldman, et al. 1963. *Computers and thought*. New York McGraw-Hill.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Danqing Huang, Jing Liu, Chin-Yew Lin, and Jian Yin. 2018. Neural math word problem solver with reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 213–223.

Danqing Huang, Shuming Shi, Chin-Yew Lin, and Jian Yin. 2017. Learning fine-grained expressions to solve math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 805–814.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281.

Jierui Li, Lei Wang, Jipeng Zhang, Yan Wang, Bing Tian Dai, and Dongxiang Zhang. 2019. Modeling intra-relation in math word problems with different functional multi-head attentions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6162–6167.

Shucheng Li, Lingfei Wu, Shiwei Feng, Fangli Xu, Fengyuan Xu, and Sheng Zhong. 2020. Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2841–2852, Online. Association for Computational Linguistics.

Chao-Chun Liang, Shih-Hong Tsai, Ting-Yun Chang, Yi-Chung Lin, and Keh-Yih Su. 2016. A meaning-based English math word problem solver with understanding, reasoning and explanation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 151–155, Osaka, Japan. The COLING 2016 Organizing Committee.

Chao-Chun Liang, Yu-Shiang Wong, Yi-Chung Lin, and Keh-Yih Su. 2018. A meaning-based statistical English math word problem solver. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 652–662, New Orleans, Louisiana. Association for Computational Linguistics.

Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. Tree-structured decoding for solving math word problems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2370–2379.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, and Chitta Baral. 2020. Towards question format independent numerical reasoning: A set of prerequisite tasks. *arXiv preprint arXiv:2005.08516*.

Arindam Mitra and Chitta Baral. 2016. Learning to use formulas to solve simple arithmetic problems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2144–2153.

Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. Abstract syntax networks for code generation and semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149, Vancouver, Canada. Association for Computational Linguistics.

Subhro Roy and Dan Roth. 2018. Mapping to declarative knowledge for word problem solving. *Transactions of the Association for Computational Linguistics*, 6:159–172.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1142.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *International Conference on Learning Representations*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Qinzhuo Wu, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2020. A knowledge-aware sequence-to-tree network for math word problem solving. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7137–7146.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pages 5299–5305.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada. Association for Computational Linguistics.

Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2019. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2287–2305.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2020. Graph-to-tree learning for solving math word problems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3928–3937, Online. Association for Computational Linguistics.

A Data Preprocessing

In this section, we describe the data preprocessing steps required for our S2G model.

A.1 Converting to prefix notation

To perform top-down tree decoding, we follow (Xie and Sun, 2019) to convert the equations into their *prefix notation*, where the operators are placed in front of their operands, rather than in between. In this way, the order of the equation tokens would match the order of decoding. In our case, we also need to consider the formulas used in the equation. For a formula in the form "$F(arg1, arg2)$", we turn it into "$[F, arg1, arg2]$" so that it can fit into the prefix notation. Table 5 shows an example of this infix-to-prefix conversion for an equation with formulas.

Problem: The outer radius and inner radius of a circular annulus are 5m and 3m respectively. Find the area of this circular annulus.
Equation: x = circle_area(5) - circle_area(3) **Prefix form:** [-, circle_area, 5, circle_area, 3]

Table 5: Infix-to-prefix conversion for an equation with formulas.

A.2 Vocabulary

We follow the canonical sequence-to-sequence architecture (Sutskever et al., 2014) to prepare for the source vocabulary. For the target vocabulary, however, we have to take into consideration the way that humans solve MWPs. To solve a math problem, we use the numbers from the problem text (a dynamic vocabulary) and the mathematical operators learned before (a static vocabulary) and try to compose them into an equation. Sometimes, we also need to use external constant numbers (a static vocabulary) that are not in the problem text but would appear in the equation (e.g., 1, 2, or 3.14). These three types of vocabulary make up the vocabulary for the equations in arithmetic problems (equation 13).

$$V_{arith} = V_{number} \cup V_{op} \cup V_{const} \quad (13)$$

We follow (Xie and Sun, 2019) to use a copy mechanism (Gu et al., 2016) to copy the numbers from the problem text. Hence, we can dynamically get the problem numbers during decoding. Besides, we extend the original vocabulary by adding domain-specific formulas into it so that the decoder can generate formulas during decoding (equation 14). Table 6 shows the overall vocabulary that we use for our decoder.

Vocab Type	Instances
Operator	+, -, *, /, ^
Number	$\langle N0 \rangle$, $\langle N1 \rangle$, $\langle N2 \rangle$, ...
Constant	1, 2, 3.14
*Formula	circle_area, square_area, rectangle_perimeter, and so on.

Table 6: Types of the vocabulary.

$$V_{target} = V_{number} \cup V_{op} \cup V_{const} \cup V_{formula} \quad (14)$$

B GeometryQA

Table 7 shows the 11 formulas used for annotation.

Name	Formula	# args
Square		
square_area	side * side	1
square_perimeter	4 * side	1
Cubic		
cubic_volume	side*side*side	1
Circle		
circle_area	π * radius^2	1
circumference_r	2 * π * radius	1
circumference_d	π * diameter	1
Triangle		
triangle_area	base*height / 2	2
Rectangle		
rectangle_area	length * width	2
rectangle_perimeter	2 * (l+w)	2
Cuboid		
cuboid_volume	l* w* height	3
cuboid_surface	2*(l*w+w*h+l*h)	3

Table 7: Eleven geometry formulas used in annotating GeometryQA.

C Related Work

In this section, we briefly introduce the progress of MWP solvers, and then we focus on topics that are

closer to our work, including seq2tree solvers and knowledge graphs for problem solving.

C.1 Math Word Problem Solving

Ever since 1960s, efforts have been made to build automatic math word problem solving systems (Feigenbaum et al., 1963; Bobrow, 1964). Statistical solvers learn to map problem features into corresponding equation templates or operations to solve the problem (Kushman et al., 2014; Hosseini et al., 2014; Mitra and Baral, 2016; Liang et al., 2016, 2018; Roy and Roth, 2018). For example, Kushman et al. (2014) propose to align MWPs to their templates, while Hosseini et al. (2014) propose to find the operations by verb categorization. Semantic parsing approaches, on the other hand, parse the problem into intermediate representations using semantic parsers (Shi et al., 2015; Koncel-Kedziorski et al., 2015; Huang et al., 2017).

Recently, neural architectures have emerged as a dominant paradigm in math word problem solving. Wang et al. (2017) first attempt to use a seq2seq solver that utilize encoder-decoder architectures to encode the problem text and then decode into equations in a way similar to machine translation. Copy mechanism (Huang et al., 2018) or attention mechanisms (Li et al., 2019) are introduced to improvement the performance of seq2seq models. These seq2seq models, however, suffer from producing invalid equations, like a binary operator with three operands, because there is no grammatical constraint on its sequential decoding. To solve this problem, seq2tree models are proposed to bring into the grammatical constraints (Xie and Sun, 2019; Liu et al., 2019). We will give a more detailed introduction to seq2tree models in Section C.2.

C.2 Sequence-to-Tree Models

To convert text into structured representations, several research strands have utilized sequence-to-tree models. Dong and Lapata (2016) first use seq2tree on semantic parsing to translate text into structured logical forms. Similar frameworks are also adopted for code generation (Yin and Neubig, 2017; Rabinovich et al., 2017) where they translate code snippets into executable representations or abstract syntax trees (ASTs).

Inspired by their ideas, MWP solving also adopts seq2tree to map the problem text into expression trees. This introduces a constraint that the non-leaf nodes of the tree should be operators and leaf nodes be numbers, and thus the resulted equations are always guaranteed to be valid. Most seq2tree solvers choose bidirectional LSTM or GRU as their text encoder and use two separate models to predict the left and right nodes during decoding respectively (Xie and Sun, 2019; Zhang et al., 2020; Wu et al., 2020; Li et al., 2020). Our model differs from the previous that we use a single RNN-based decoder to predict a variable number of children nodes during decoding. In addition, our model can predict formulas as nodes that increase the interpretability of the model outputs, while previous solvers can only predict basic arithmetic operators.

C.3 Knowledge Graph for Math Word Problem Solving

To incorporate external knowledge into problem solving, some solvers utilize graph convolutional networks (Kipf and Welling, 2017) or graph attention networks (Veličković et al., 2018) to encode knowledge graphs (KGs) as an additional input. Wu et al. (2020) proposed to incorporate commonsense knowledge from external knowledge bases. They constructed a dynamic KG for each problem to model the relationship between the entities in the problem. For example, ”daisy” and ”rose” would be linked to their category ”flower” so that the solver can use this hyperonymy information when counting the number of flowers. Zhang et al. (2020) proposed to build graphs that model the quantity-related information using dependency parsing and POS tagging tools (Manning et al., 2014). Their graphs provide better quantity representations to the solver. Our model differs from previous models that we aim to incorporate domain knowledge from mathematical KGs rather than from commonsense knowledge bases.

Multi-Scale Progressive Attention Network for Video Question Answering

Zhicheng Guo Jiaxuan Zhao Licheng Jiao Xu Liu Lingling Li
School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi Province, 710071, China
{zchguo, jiaxuanzhao}@stu.xidian.edu.cn
lchjiao@mail.xidian.edu.cn {xuliu, llli}@xidian.edu.cn

Abstract

Understanding the multi-scale visual information in a video is essential for Video Question Answering (VideoQA). Therefore, we propose a novel Multi-Scale Progressive Attention Network (MSPAN) to achieve relational reasoning between cross-scale video information. We construct clips of different lengths to represent different scales of the video. Then, the clip-level features are aggregated into node features by using max-pool, and a graph is generated for each scale of clips. For cross-scale feature interaction, we design a message passing strategy between adjacent scale graphs, i.e., top-down scale interaction and bottom-up scale interaction. Under the question's guidance of progressive attention, we realize the fusion of all-scale video features. Experimental evaluations on three benchmarks: TGIF-QA, MSVD-QA and MSRVTT-QA show our method has achieved state-of-the-art performance.

1 Introduction

Video Question Answering (VideoQA) is a popular vision-language task, which focuses on predicting the correct answer to a given natural language question based on the corresponding video. VideoQA task entails representing video features in both spatial and temporal dimensions. Compared with the visual features of a picture in Visual Question Answering, it requires a more complex attention.

Therefore, (Jang et al., 2017) employed appearance features and motion features as video representation, and designed a dual-LSTM network based on spatio-temporal attention to fuse visual and text information. Next, memory networks are widely used to capture long-term dependencies. For example, (Cai et al., 2020) applied feature augmented memory to strengthen the information augmentation of video and text. Complex relational reasoning is important for VideoQA task. Consequently, a conditional relationship network (Le et al., 2020) was designed in previous work, which can support high-order relationships and multi-step reasoning.

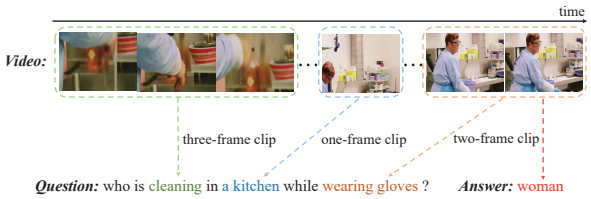


Figure 1: Understanding the video and answering the question require different levels of clips.

Many methods complete this task from a certain aspect, however, none of them have a fine-grained understanding of video information. When looking for the answer in a question-based video, the video frames corresponding to different objects in the question are of different lengths. As shown in Fig. 1, when asked "who is cleaning in a kitchen while wearing gloves?", we need to find the keywords "cleaning", "a kitchen" and "wearing gloves" from different levels of clips. Previous methods searched for the answer on the same level of clips in a video, leading to insufficient or redundant information.

Firstly, we construct clips of different lengths from the frame sequence, and regard the length of a clip as its scale information. Then, multi-scale graphs are generated separately for clips of different scales. The nodes in the multi-scale graphs indicate video features corresponding to different clips. For implementing relational reasoning, the nodes in each scale graph are first updated by using graph convolution. Most importantly, under the guidance of the question, progressive attention has been utilized to enable the fusion of multi-scale features during cross-scale graph interaction. In detail, each graph is gradually updated in top-down scale order, followed by updating each graph in bottom-up scale order. Finally, node features of a graph are fused with question embedding, and a classifier is employed to find the answer.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 973–978
August 1–6, 2021. ©2021 Association for Computational Linguistics

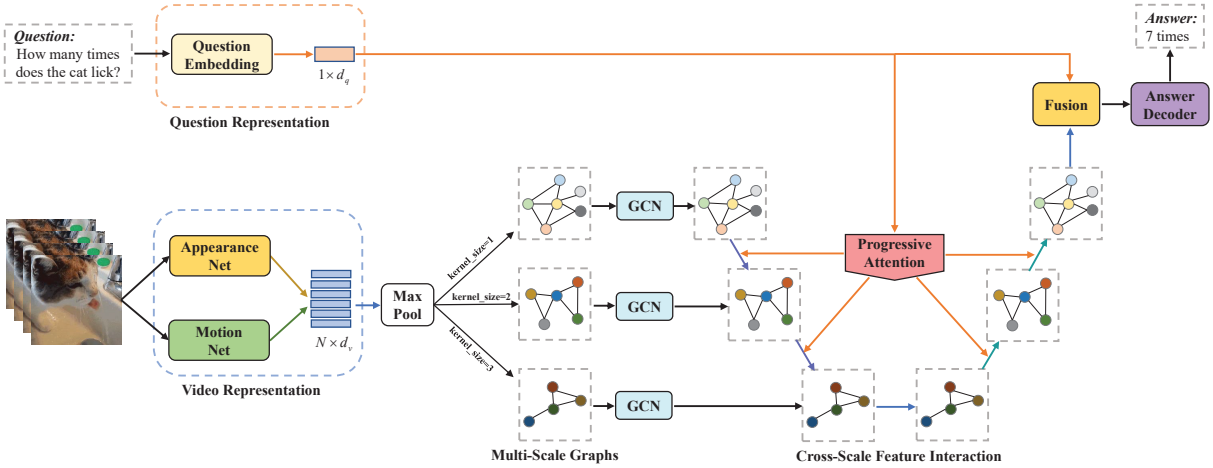


Figure 2: The model architecture of Multi-Scale Progressive Attention Network for VideoQA. Major contributions focus on the construction of multi-scale graphs and the progressive attention for cross-scale feature interaction.

2 Method

An overview of the proposed MSPAN is shown in Fig. 2. The input is a short video and a question sentence, while the output is the produced answer.

2.1 Video and Question Representation

Video representation N frames are uniformly sampled to represent the video. Then we use the pre-trained ResNet-152 (He et al., 2016) to extract video appearance features for each frame. And, we apply the 3D ResNet-152 (Hara et al., 2018) pre-trained on Kinetics-700 (Carreira et al., 2019) dataset to extract video motion features. Specifically, 16 frames around each frame are placed into the 3D ResNet-152 to obtain the motion features around this frame. Finally, we get a joint video representation by concatenating appearance features and motion features. By using a fully-connected layer to reduce feature dimension, we obtain video representation as $V = \{v_i : i \leq N, v_i \in R^{2048}\}$.

Question representation All words in question are represented as 300-dimensional embeddings initialized with pre-trained GloVe vectors (Pennington et al., 2014). And a 512-dimensional question embedding is generated from the last hidden state of a three-layer BiLSTM, i.e., $q \in R^{512}$.

2.2 Multi-Scale Graphs Generation

Each object in the video corresponds to a different number of frames, but previous methods (Seo et al., 2020; Lei et al., 2021) cannot treat various levels of visual information separately. Therefore, we construct clips of different lengths to express the visual information in the video delicately, and regard the length attribute as a scale.

We use max-pools of different kernel-sizes to aggregate frame-level visual features, and kernel-size is the scale attribute of these clips. In this way, clip-level visual features are obtained, as follows:

$$P = \{pool_i | 1 \leq i \leq K, kernel\_size_i = i\} \quad (1)$$

$$V_i = P_i(v_1, v_2, \cdots, v_N) \quad (2)$$

Where K is the range of scales, and $K \leq N$. Thus, we construct $M_i = N - i + 1$ clips at scale i:

$$V_i = \{v_j^i : 1 \leq j \leq M_i, v_j^i \in R^{2048}\} \quad (3)$$

In order to reason the relationships between different objects in a video, we separately build a graph for each scale. Each node in a graph represents the clip-level visual features. Only when two nodes contain overlapping or adjacent frames, an edge will be connected between them. Frame interval of the j-th clip at scale i is $[j, j + i - 1]$, so all edges in the K graphs can be expressed as:

$$E_i = \{(x, y) | x - i \leq y \leq x + i\} \quad (4)$$

Finally, these multi-scale graphs constructed in this paper can be denoted as $G_i = \{V_i, E_i\}$.

2.3 Cross-Scale Feature Interaction

Before cross-scale feature interaction, the original node features of K graphs are copied as $V_i^o = V_i$.

Interaction at the same scale. For all nodes with the same scale, we apply a two-layer graph convolutional network (GCN) (Kipf and Welling, 2017) to perform relational reasoning over the K

graphs. The process of graph convolution is represented as:

$$X_{l+1} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}X_l W_l \quad (5)$$

Where $\hat{A}$ is the input adjacency matrix, X_l is the node feature matrix of layer l, and W_l is the learnable weight matrix. The diagonal node degree matrix $\hat{D}$ is used to normalize $\hat{A}$. Due to the small number of nodes in each graph, we decide to share the weight matrix W_l when K graphs are updated.

Interaction at top-down scale. We realize the interaction of adjacent scale graphs from small scale to large scale. Therefore, visual information is understood step by step from details to the whole through the interaction of top-down scale. Guided by the question, the nodes in graph G_i are used to update the nodes in graph G_{i+1}. Visual features at different scales show hierarchical attention to the question, so we call it progressive attention.

If the clip corresponding to node x in graph G_i has the same frames as the clip corresponding to node y in graph G_{i+1}, there will exist a directed edge from x to y. Therefore, we can use the edge to fuse the cross-scale features of these same frames.

Firstly, visual features and question embedding are fused to capture the joint features of each node in graph G_i. Then, the process of message passing from graph G_i to graph G_{i+1} can be expressed as:

$$m_{xy} = (W_1 v_y^{i+1}) \otimes ((W_2 v_x^i) \odot (W_3 q))^T \quad (6)$$

Where $\otimes$ is the outer product, $\odot$ is the hadamard product. After receiving the delivery messages, the attention weights of these messages are calculated:

$$w_{xy} = \underset{x \in \mathcal{N}_y}{soft\max}(m_{xy}) \quad (7)$$

Where $\mathcal{N}_y$ is the set of all neighbor nodes in graph G_i through cross-scale edges. Consequently, all the messages passed into node y are summed to derive the update of node y, as follows:

$$\tilde{v}_y^{i+1} = \sum_{x \in N_y} w_{xy} \cdot ((W_4 v_x^i) \odot (W_5 q)) \quad (8)$$

$$V_{i+1}^u = \{\tilde{v}_y^{i+1} : y \leq M_{i+1}, \tilde{v}_y^{i+1} \in R^{2048}\} \quad (9)$$

When updating all nodes in graph G_{i+1}, we consider the new features V_{i+1}^u and the original features V_{i+1}^o. Therefore, we use the residual connection to preserve original information of the video:

$$V_{i+1} = W_6[V_{i+1}; V_{i+1}^u] + V_{i+1}^o \quad (10)$$

Where $[;]$ is the concatenation operator. Above $W_1 \sim W_6$ are learnable weights, and they are shared in the update of graphs $G_2 \sim G_K$. To summarize, the update of $K-1$ graphs is a progressive process from small scale to large scale, hence it is referred to as top-down scale interaction.

Interaction at bottom-up scale. After an overall understanding of the video, people can accurately find all details related to the question at the second time they watch the video. Therefore, we achieve an understanding of the video from global to local through bottom-up scale interaction. After the previous interaction, we realize the interaction of adjacent graphs from large scale to small scale.

Following the same method as top-down scale interaction from Eq. 6 to Eq. 10, we apply graph G_i to update graph G_{i-1} in this interaction. But the weights $W_1 \sim W_6$ are another group in the update of graphs $G_{K-1} \sim G_1$. After this interaction, graph G_1 can grasp the all-scale video features related to the question by progressive attention.

2.4 Multimodal Fusion and Answer Decoder

After T iterations of cross-scale feature interaction, we read out all the nodes in graph G_1. Then, a simple attention is used to aggregate the N nodes. And, final multi-modal representation is given as:

$$\tilde{w}_j = soft\max(W_7(W_8 v_j^1) \odot (W_9 q)) \quad (11)$$

$$\tilde{F} = \sum_{j=1}^{N} \tilde{w}_j \cdot v_j^1 \quad (12)$$

$$F = ELU(W_{10}\tilde{F} \odot W_{11} q + b) \quad (13)$$

Where ELU is activation function, above $W_7 \sim W_{11}$ are learnable weights and b is learnable bias. We can find the answer by applying a classifier (two fully-connected layers) on multi-modal representation F. Multi-label classifier is applied to open-ended tasks, and cross-entropy loss function is used to train the model. Due to repetition count is a regression task, we use the MSE loss function.

For the multi-choice task, each question corresponds to R answer sentences. We first get the embedding of each answer in the same way as the question embedding. Then we use the multi-modal fusion method in Eq. 11∼13 to fuse the answer embedding with node features. After using two fully-connected layers, the answer scores $\{s_i\}_{i=1}^R$ have appeared. This model is trained by minimizing the hinge loss (Jang et al., 2017) of pairwise comparisons between answer scores $\{s_i\}_{i=1}^R$.

3 Experiments

3.1 Datasets

TGIF-QA (Jang et al., 2017) is a widely used large-scale benchmark dataset for VideoQA. And four task types are covered in this dataset: repeating action (Action), repetition count (Count), video frame QA (FrameQA) and state transition (Trans.). **MSVD-QA** (Xu et al., 2017) and **MSRVTT-QA** (Xu et al., 2016) are open-ended tasks which are generated from video descriptions. In both datasets, questions can be divided into 5 types according to question words: what, who, how, when and where.

3.2 Implementation Details

We evenly sample $N = 16$ frames for each video in the three datasets. The hyperparameters we set in experiments are as follows: $T = 3$, $K = 8$. When training the network, Adam is used with an initial learning rate of 10^{-4}. For TGIF-QA dataset, the batch size is 64. While the batch size is set to 128 for both MSVD-QA and MSRVTT-QA datasets.

3.3 Results

We compare our MSPAN with the state-of-the-art methods: PSAC (Li et al., 2019), HME (Fan et al., 2019), FAM (Cai et al., 2020), LGCN (Huang et al., 2020), HGA (Jiang and Han, 2020), QueST (Jiang et al., 2020) and HCRN (Le et al., 2020).

Table 1: Comparison on TGIF-QA dataset.

Method	Action	Count	FrameQA	Trans.
PSAC	70.4	4.27	55.7	76.9
HME	73.9	4.02	53.8	77.8
FAM	75.4	3.79	56.9	79.2
LGCN	74.3	3.95	56.3	81.1
HGA	75.4	4.09	55.1	81.0
QueST	75.9	4.19	59.7	81.0
HCRN	75.0	3.82	55.9	81.4
MSPAN	**78.4**	**3.57**	**59.7**	**83.3**

Results on TGIF-QA. As shown in Table 1, our method outperforms the state-of-the-art methods by **2.5%** and **1.9%** of accuracy on Action and Transition tasks. For the Count task, our method also achieves the best Mean Square Error (MSE) of **3.57** among all methods. Due to QueST used multi-dimension visual features containing more appearance information, our method can only get the same accuracy **59.7%** as QueST on the FrameQA task.

Table 2: Comparison on MSVD-QA dataset.

Method	What	Who	How	When	Where	All
	62.7%	33.9%	2.8%	0.4%	0.2%	100%
HME	22.4	50.1	73.0	70.7	42.9	33.7
QueST	24.5	52.9	79.1	72.4	50.0	36.1
HGA	23.5	50.4	**83.0**	72.4	46.4	34.7
FAM	23.1	51.6	82.2	71.4	51.9	34.5
MSPAN	**31.0**	**53.8**	77.0	**72.4**	**53.6**	**40.3**

Table 3: Comparison on MSRVTT-QA dataset.

Method	What	Who	How	When	Where	All
	68.5%	27.7%	2.5%	1.0%	0.3%	100%
HME	26.5	43.6	82.4	76.0	28.6	33.0
QueST	27.9	45.6	83.0	75.7	31.6	34.6
HGA	29.2	45.7	**83.5**	75.2	34.0	35.5
FAM	26.9	43.9	82.8	70.6	31.1	33.2
MSPAN	**31.9**	**47.2**	83.2	**77.5**	**38.4**	**37.8**

All in all, our method makes sense of the multi-scale information of the video, so that the effect on tasks related to action recognition, temporal relationship and object count are very noticeable.

Results on MSVD-QA. As shown in Table 2, our method improves the overall accuracy by **4.2%** compared to recent methods. We have achieved the best accuracy on questions whose question words are "What" , "Who", "When" and "Where". Due to a small proportion, the accuracy on the question word "How" is lower than other methods.

Results on MSRVTT-QA. As shown in Table 3, our method achieves the best overall accuracy of **37.8%**. What's more, Our method could obtain excellent accuracy on different question words.

4 Ablation Studies

To explore the potential of our network, ablation experiments are performed on TGIF-QA dataset. Default hyperparameters are: $T = 3$ and $K = 8$. We study the effectiveness of our network in the next two aspects, as shown in Table 4 and Fig. 4.

4.1 Different Structures

Considering the interaction of cross-scale graphs, three structures are designed, as shown in Fig. 3. For the dense scale in Fig. 3 (a), we apply graphs $G_1 \sim G_K$ to update each graph G_i. The other two structures have been introduced in Sec 2.3, and we will not use a graph to update itself for the three

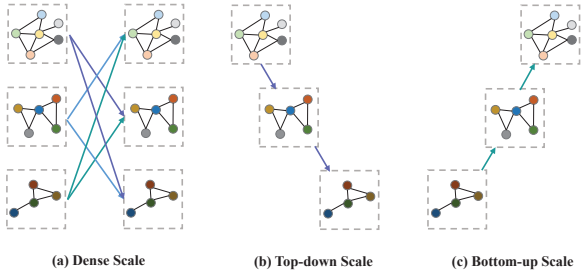


Figure 3: Three methods of cross-scale feature interaction, where the dense connection is not adopted.

structures. The readout of top-down scale interaction is graph G_K, and the readout of bottom-up scale interaction is G_1. However, the readout of dense scale interaction is all K graphs. Our network is a combination of top-down scale interaction and bottom-up scale interaction, but we will use these two structures separately for comparison.

4.2 Network structure

When choosing the pooling function to aggregate these frames in a clip, we find that max-pool is more effective than avg-pool. In reverse gradient propagation of max-pool, only the maximum of features in the previous layer receive the gradient. So, max-pool facilitates the fusion of appearance features and motion features in the previous layer.

Our experiments show that GCN is beneficial to the stable training of models. If there is no GCN, the gradient will gradually disappear as the number of interactions between the graphs increases. The role of GCN is to re-recover the features of these nodes which have lost their visual features.

As shown in Table 4, the performances of the three structures in Fig. 3 are poorer than that of our entire network. Due to dense connections between all scale graphs, the dense scale interaction will add much unnecessary computation, and make it difficult to accurately find the visual information related to the question. Although both the top-down scale interaction and the bottom-up scale interaction can achieve good performance. However, the combination of these two structures will obtain a more detailed understanding of the video.

4.3 Hyperparameters T and K

As the number of iterations T increases, the model will achieve better performance. But when $T = 4$, the effect of the model decreases, as shown in Table 4. Because too many modules will produce noise for answer generation. The improvement

Table 4: Ablation experiments of four types: (1)Replacing max-pool with avg-pool. (2)Without GCN. (3)Different structures in Fig. 3. (4)Different iterations T.

Parameters	Action	Count	FrameQA	Trans.
Avg-pool	78.0	3.56	59.5	83.3
w/o GCN	77.5	3.64	59.1	82.7
Dense scale	77.2	3.74	59.2	82.0
Top-down scale	78.1	3.62	59.6	82.8
Bottom-up scale	78.1	3.60	59.3	82.6
$T = 0$	75.2	3.86	56.7	79.9
$T = 1$	77.1	3.69	58.6	82.5
$T = 2$	77.7	3.61	**59.7**	82.9
$T = 4$	77.6	3.63	59.4	82.5
Full MSPAN	**78.4**	**3.57**	59.7	**83.3**

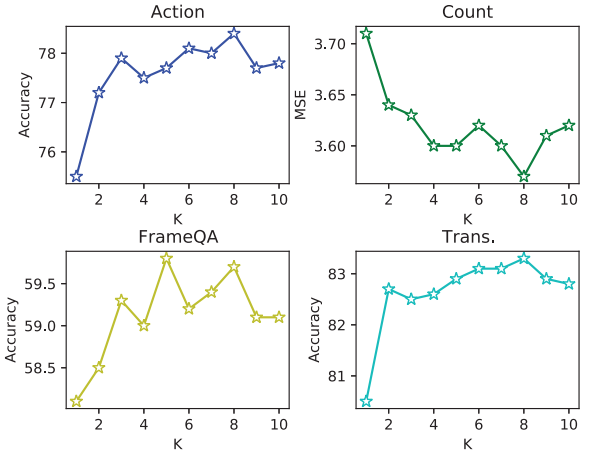


Figure 4: Ablation experiments for different scales K.

of models with the increase of K is very obvious, and best performance is obtained when $K = 8$, as shown in Fig. 4. However, the larger K also means that many multi-scale graphs, which will lead to network instability.

5 Conclusion

We introduce a multi-scale learning method to achieve a fine-grained understanding of the video. Compared with existing spatio-temporal attention, we use progressive attention to realize cross-scale feature interaction. The top-down and bottom-up structures we have designed are conducive to learning all-scale visual information of the video. For longer videos, we plan to use dilated max-pools with different strides to reduce the size of graphs. In general, we consider the VideoQA task from the perspective of multi-scale information interaction, and the proposed network is instructive.

References

Jiayin Cai, Chun Yuan, Cheng Shi, Lei Li, Yangyang Cheng, and Ying Shan. 2020. Feature augmented memory with global attention network for videoqa. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 998–1004.

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. 2019. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*.

Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. 2019. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1999–2007.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. 2020. Location-aware graph convolutional networks for video question answering. In *AAAI*, pages 11021–11028.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2758–2766.

Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. 2020. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *AAAI*, pages 11101–11108.

Pin Jiang and Yahong Han. 2020. Reasoning with heterogeneous graph alignment for video question answering. In *AAAI*, pages 11109–11116.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2020. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9972–9981.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. *arXiv preprint arXiv:2102.06183*.

Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. 2019. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8658–8665.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. 2020. Look before you speak: Visually contextualized utterances. *arXiv preprint arXiv:2012.05710*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Efficient Passage Retrieval with Hashing for Open-domain Question Answering

Ikuya Yamada[†,‡] **Akari Asai**[*] **Hannaneh Hajishirzi**[*,§]
[†]Studio Ousia [‡]RIKEN [*]University of Washington
[§]Allen Institute for AI
ikuya@ousia.jp {akari,hannaneh}@cs.washington.edu

Abstract

Most state-of-the-art open-domain question answering systems use a neural retrieval model to encode passages into continuous vectors and extract them from a knowledge source. However, such retrieval models often require large memory to run because of the massive size of their passage index. In this paper, we introduce *Binary Passage Retriever* (*BPR*), a memory-efficient neural retrieval model that integrates a *learning-to-hash* technique into the state-of-the-art Dense Passage Retriever (DPR) (Karpukhin et al., 2020) to represent the passage index using compact binary codes rather than continuous vectors. BPR is trained with a multi-task objective over two tasks: efficient candidate generation based on binary codes and accurate reranking based on continuous vectors. Compared with DPR, BPR substantially reduces the memory cost from 65GB to 2GB without a loss of accuracy on two standard open-domain question answering benchmarks: Natural Questions and TriviaQA. Our code and trained models are available at https://github.com/studio-ousia/bpr.

1 Introduction

Open-domain question answering (QA) is the task of answering arbitrary factoid questions based on a knowledge source (e.g., Wikipedia). Recent state-of-the-art QA models are typically based on a two-stage *retriever–reader* approach (Chen et al., 2017) using a *retriever* that obtains a small number of relevant passages from a large knowledge source and a *reader* that processes these passages to produce an answer. Most recent successful retrievers encode questions and passages into a common continuous embedding space using two independent encoders (Lee et al., 2019; Karpukhin et al., 2020; Guu et al., 2020). Relevant passages are retrieved using a nearest neighbor search on the index containing the passage embeddings with a question embedding as a query.

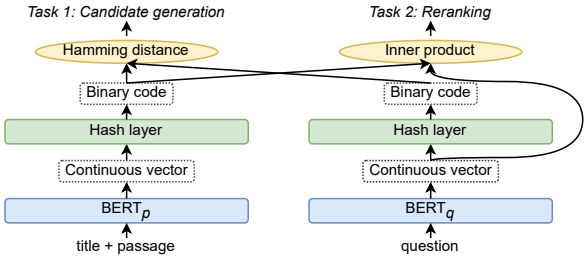


Figure 1: Architecture of BPR, a BERT-based model generating compact binary codes for questions and passages. The passages are retrieved in two stages: (1) efficient candidate generation based on the Hamming distance using the binary code of the question and (2) accurate reranking based on the inner product using the continuous embedding of the question.

These retrievers often outperform classical methods (e.g., BM25), but they incur a large memory cost due to the massive size of their passage index, which must be stored entirely in memory at runtime. For example, the index of a common knowledge source (e.g., Wikipedia) requires dozens of gigabytes.[1] A reduction in the index size is critical for real-world QA that requires large knowledge sources such as scientific databases (e.g., PubMed) and web-scale corpora (e.g., Common Crawl).

In this paper, we introduce *Binary Passage Retriever* (*BPR*), which learns to hash continuous vectors into compact binary codes using a multi-task objective that simultaneously trains the encoders and hash functions in an end-to-end manner (see Figure 1). In particular, BPR integrates our *learning-to-hash* technique into the state-of-the-art Dense Passage Retriever (DPR) (Karpukhin et al., 2020) to drastically reduce the size of the

[1]The passage index of the off-the-shelf DPR model (Karpukhin et al., 2020) requires 65GB for indexing the 21M English Wikipedia passages, which have 13GB storage size.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 979–986
August 1–6, 2021. ©2021 Association for Computational Linguistics

passage index by storing it as binary codes. BPR computes binary codes by applying the sign function to continuous vectors. As the sign function is not compatible with back-propagation, we approximate it using the scaled tanh function during training. To improve search-time efficiency while maintaining accuracy, BPR is trained to obtain both binary codes and continuous embeddings for questions with multi-task learning over two tasks: (1) candidate generation based on the Hamming distance using the binary code of the question and (2) reranking based on the inner product using the continuous embedding of the question. The former task aims to detect a small number of candidate passages efficiently from the entire passages and the latter aims to rerank the candidate passages accurately.

We conduct experiments using the Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (TQA) (Joshi et al., 2017) datasets. Compared with DPR, our BPR achieves similar QA accuracy and competitive retrieval performance with a substantially reduced memory cost from 65GB to 2GB. Furthermore, using an improved reader, we achieve results that are competitive with those of the current state of the art in open-domain QA. Our code and trained models are available at `https://github.com/studio-ousia/bpr`.

2 Related Work

Retrieval for Open-domain QA Many recent open-domain QA models depend on the retriever to select relevant passages from a knowledge source. Early works involved the adoption of sparse representations (Chen et al., 2017) for the retriever, whereas recent works (Lee et al., 2019; Guu et al., 2020; Karpukhin et al., 2020) have often adopted dense representations based on neural networks. Our work is an extension of DPR (Karpukhin et al., 2020), which has been used in recent state-of-the-art QA models (Lewis et al., 2020; Izacard and Grave, 2020). Concurrent with our work, Izacard et al. (2020) attempted to reduce the memory cost of DPR using post-hoc product quantization with dimension reduction and filtering of passages. However, they observed a significant degradation in the QA accuracy compared with their full model. We adopt the learning-to-hash method with our multi-task objective and substantially compress the index without losing accuracy.

Learning to Hash The objective of hashing is to reduce the memory and search-time cost of the nearest neighbor search by representing data points using compact binary codes. Learning to hash (Wang et al., 2016, 2018) is a method for learning hash functions in a data-dependent manner. Recently, many *deep-learning-to-hash* methods have been proposed (Lai et al., 2015; Zhu et al., 2016; Li et al., 2016; Cao et al., 2017, 2018) to jointly learn feature representations and hash functions in an end-to-end manner. We follow Cao et al. (2017) to implement our hash functions. Similar to our work, Xu and Li (2020) used the learning-to-hash method to reduce the computational cost of the answer sentence selection task, the objective of which is to select an answer sentence from a limited number of candidates (up to 500 in their experiments). Our work is different from the aforementioned work because we focus on efficient and scalable passage retrieval from a large knowledge source (21M Wikipedia passages in our experiments) using an effective multi-task approach. In addition to hashing-based methods, improving approximate neighbor search has been actively studied (Jégou et al., 2011; Malkov and Yashunin, 2020; Guo et al., 2020). We use Jégou et al. (2011) and Malkov and Yashunin (2020) as baselines in our experiments.

3 Model

Given a question and large-scale passage collection such as Wikipedia, a retriever finds relevant passages that are subsequently processed by a reader. Our retriever is built on DPR (Karpukhin et al., 2020), which is a retriever based on BERT (Devlin et al., 2019). In this section, we first introduce DPR and then explain our model.

3.1 Dense Passage Retriever (DPR)

DPR uses two independent BERT encoders to encode question q and passage p into d-dimensional continuous embeddings:

$$\mathbf{e}_q = \mathrm{BERT}_q(q), \quad \mathbf{e}_p = \mathrm{BERT}_p(p), \qquad (1)$$

where $\mathbf{e}_q \in \mathbb{R}^d$ and $\mathbf{e}_p \in \mathbb{R}^d$. We use the uncased version of BERT-base; therefore, $d = 768$. The output representation of the `[CLS]` token is obtained from the encoder. To create passage p, the passage title and text are concatenated (`[CLS]` *title* `[SEP]` *passage* `[SEP]`). The relevance score of passage p, given question q, is computed using the inner product of the corresponding vectors, $\langle \mathbf{e}_q, \mathbf{e}_p \rangle$.

Training Let $\mathcal{D} = \{\langle q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^- \rangle\}_{i=1}^m$ be m training instances consisting of a question q_i, a passage that answers the question (positive passage), p_i^+, and n passages that are irrelevant for the question (negative passages), $p_{i,j}^-$. The model is trained by minimizing the negative log-likelihood of the positive passage:

$$\mathcal{L}_{\text{dpr}} = -\log \frac{\exp(\langle \mathbf{e}_{q_i}, \mathbf{e}_{p_i^+} \rangle)}{\exp(\langle \mathbf{e}_{q_i}, \mathbf{e}_{p_i^+} \rangle) + \sum_{j=1}^n \exp(\langle \mathbf{e}_{q_i}, \mathbf{e}_{p_{i,j}^-} \rangle)}. \quad (2)$$

Inference DPR creates a passage index by applying the passage encoder to each passage in the knowledge source. At runtime, it retrieves the top-k passages employing maximum inner product search with the question embedding as a query.

3.2 Model Architecture

Figure 1 shows the architecture of BPR. BPR builds a passage index by computing a binary code for each passage in the knowledge source. To compute the binary codes for questions and passages, we add a *hash layer* on top of the question and passage encoders of DPR. Given embedding $\mathbf{e} \in \mathbb{R}^d$ computed by an encoder, the hash layer computes its binary code, $\mathbf{h} \in \{-1, 1\}^d$, as

$$\mathbf{h} = \text{sign}(\mathbf{e}), \quad (3)$$

where $\text{sign}(\cdot)$ is the sign function such that for $i = 1, ..., d$, $\text{sign}(h_i) = 1$ if $h_i > 0$; otherwise, $\text{sign}(h_i) = -1$. However, the sign function is incompatible with back-propagation because its gradient is zero for all non-zero inputs and is ill-defined at zero. Inspired by Cao et al. (2017), we address this by approximating the sign function using the scaled tanh function during the training:

$$\tilde{\mathbf{h}} = \tanh(\beta \mathbf{e}), \quad (4)$$

where β is a scaling parameter. When β increases, the function gradually becomes non-smooth, and as $\beta \to \infty$, it converges to the sign function. At each training step, we increase β by setting $\beta = \sqrt{\gamma \cdot step + 1}$, where $step$ is the number of finished training steps. We set $\gamma = 0.1$ and explain the effects of changing it in Appendix B.

3.3 Two-stage Approach

To reduce the computational cost without losing accuracy, BPR splits the task into candidate generation and reranking stages. At the candidate generation stage, we efficiently obtain the top-l candidate passages using the Hamming distance between the binary code of question $\mathbf{h}_q$ and that of each passage, $\mathbf{h}_p$. We then rerank the l candidate passages using the inner product between the continuous embedding of question $\mathbf{e}_q$ and $\mathbf{h}_p$ and select the top-k passages from the reranked candidates. We perform candidate generation using binary code $\mathbf{h}_q$ for search-time efficiency, and reranking using expressive continuous embedding $\mathbf{e}_q$ for accuracy. We set $l = 1000$ and describe the effects of using different l values in Appendix C.

3.4 Training

To compute effective representations for both the candidate generation and reranking stages, we combine the loss functions of the two tasks:

$$\mathcal{L} = \mathcal{L}_{\text{cand}} + \mathcal{L}_{\text{rerank}}. \quad (5)$$

Task #1 for Candidate Generation The objective of this task is to improve candidate generation using the ranking loss with the approximated hash code of question $\tilde{\mathbf{h}}_q$ and that of passage $\tilde{\mathbf{h}}_p$:

$$\mathcal{L}_{\text{cand}} = \sum_{j=1}^n \max(0, -(\langle \tilde{\mathbf{h}}_{q_i}, \tilde{\mathbf{h}}_{p_i^+} \rangle + \langle \tilde{\mathbf{h}}_{q_i}, \tilde{\mathbf{h}}_{p_{i,j}^-} \rangle) + \alpha). \quad (6)$$

We set $\alpha = 2$ and investigate the effects of selecting different α values and using the cross-entropy loss instead of the ranking loss in Appendix D. Note that the retrieval performance based on the Hamming distance can be optimized using this loss function because the Hamming distance and inner product can be used interchangeably for binary codes.[2]

Task #2 for Reranking We improve the reranking stage using the following loss function:

$$\mathcal{L}_{\text{rerank}} = -\log \frac{\exp(\langle \mathbf{e}_{q_i}, \tilde{\mathbf{h}}_{p_i^+} \rangle)}{\exp(\langle \mathbf{e}_{q_i}, \tilde{\mathbf{h}}_{p_i^+} \rangle) + \sum_{j=1}^n \exp(\langle \mathbf{e}_{q_i}, \tilde{\mathbf{h}}_{p_{i,j}^-} \rangle)}. \quad (7)$$

This function is equivalent to $\mathcal{L}_{\text{dpr}}$, with the exception that $\tilde{\mathbf{h}}_p$ is used instead of $\mathbf{e}_p$.

3.5 Algorithms for Candidate Generation

To perform candidate generation, we test two standard algorithms: (1) linear scan based on efficient Hamming distance computation,[3] and (2) hash table lookup implemented by building a hash table that maps each binary code to the corresponding passages and querying it multiple times by increasing the Hamming radius until we obtain l passages.

[2] Given two binary codes, $\mathbf{h}_i$ and $\mathbf{h}_j$, there exists a relationship between their Hamming distance, $\text{dist}_H(\cdot, \cdot)$, and inner product, $\langle \cdot, \cdot \rangle$: $\text{dist}_H(\mathbf{h}_i, \mathbf{h}_j) = \frac{1}{2}(const - \langle \mathbf{h}_i, \mathbf{h}_j \rangle)$.

[3] The Hamming distance can be computed more efficiently than the inner product using the POPCNT CPU instruction.

Model	Top 1		Top 20		Top 100		QA Acc. (EM)		Index	Query
	NQ	TQA	NQ	TQA	NQ	TQA	NQ	TQA	size	time
DPR	**46.0**	**53.5**	78.4	**79.4**	85.4	**85.0**	41.5	**56.8**	64.6GB	456.9ms
DPR + HNSW	45.7	53.2	**78.8**	78.8	85.2	84.2	41.2	56.6	151.0GB	1.8ms
DPR + Simple LSH	21.5	28.4	63.9	65.2	77.2	76.9	35.8	48.1	2.0GB	28.8ms
DPR + PQ	32.5	42.8	72.2	73.2	81.2	80.4	38.4	52.0	2.0GB	44.0ms
BPR (linear scan; $l = 1000$)	41.1	49.7	77.9	77.9	**85.7**	84.5	**41.6**	**56.8**	2.0GB	85.3ms
BPR (hash table lookup; $l = 1000$)	"	"	"	"	"	"	"	"	2.2GB	38.1ms

Table 1: Top k recall and exact match (EM) QA accuracy on test sets with the index size and query time of BPR and baselines. All models use the same reader based on BERT-base to evaluate the QA accuracy.

Model	Top 1		Top 20		Top 100		Query
	NQ	TQA	NQ	TQA	NQ	TQA	time
BPR ($l = 1000$)	41.1	49.7	77.9	77.9	85.7	84.5	38.1ms
BPR w/o reranking	38.0	46.1	76.5	75.9	84.9	83.4	37.9ms
BPR w/o candidate generation	41.1	49.7	77.9	77.9	85.7	84.5	457.8ms

Table 2: Results of our ablation study. Hash table lookup is used to implement candidate generation.

4 Experiments

Datasets We conduct experiments using the NQ and TQA datasets and English Wikipedia as the knowledge source. We use the following pre-processed data available on the DPR website:[4] Wikipedia corpus containing 21M passages and the training/validation datasets for the retriever containing multiple positive, *random* negative, and *hard* negative passages for each question.

Baselines We compare our BPR with DPR with linear scan and DPR with Hierarchical Navigable Small World (HSNW) graphs (Malkov and Yashunin, 2020) – which builds a multi-layer structure consisting of a hierarchical set of proximity graphs, following Karpukhin et al. (2020) – for our primary baselines. We also apply two popular post-hoc quantization algorithms to the DPR passage index: simple locality sensitive hashing (LSH) (Neyshabur and Srebro, 2015) and product quantization (PQ) (Jégou et al., 2011). We configure these algorithms such that their passage representations have the same size as that of BPR: the number of bits per passage of the LSH is set as 768, and the number of centroids and the code size of the PQ are configured as 96 and 8 bits, respectively.

Experimental settings Our experimental setup follows Karpukhin et al. (2020). We evaluate our model based on its top-k recall (the percentage of positive passages in the top-k passages), retrieval efficiency (the index size and query time), and exact match (EM) QA accuracy measured by combining our model with a reader. We use the same BERT-based reader as that used by DPR. Our model is trained using the same method as DPR. We conduct experiments on servers with two Intel Xeon E5-2698 v4 CPUs and eight Nvidia V100 GPUs. The passage index are built using Faiss (Johnson et al., 2019). Further details are provided in Appendix A.

4.1 Results

Main results Table 1 presents the top-k recall (for $k \in \{1, 20, 100\}$), EM QA accuracy, index size, and query time achieved by BPR and baselines on the NQ and TQA test sets. BPR achieves similar or even better performance than DPR in both retrieval with $k \geq 20$ and EM accuracy with a substantially reduced index size from 65GB to 2GB. We observe that BPR performs worse than DPR for $k = 1$, but usually the recall in small k is less important because the reader usually produces an answer based on $k \geq 20$ passages. BPR significantly outperforms all quantization baselines. The query time of BPR is substantially shorter than that of DPR. Hash table lookup is faster than linear scan but requires slightly more storage. DPR+HNSW is faster than BPR; however, it requires 151GB of storage.

Ablations Table 2 shows the results of our ablation study. Disabling the reranking clearly degrades performance, demonstrating the effectiveness of our two-stage approach. Disabling the can-

[4] https://github.com/facebookresearch/DPR

Model	Pretrained model	#params	NQ	TQA
RAG (Lewis et al., 2020)	BART-large	406M	44.5	56.1
FiD (base) (Izacard and Grave, 2020)	T5-base	220M	48.2	65.0
FiD (large) (Izacard and Grave, 2020)	T5-large	770M	**51.4**	**67.6**
BPR ($l = 1000$)	BERT-base	110M	41.6	56.8
BPR ($l = 1000$)	ELECTRA-large	335M	49.0	65.6

Table 3: Exact match QA accuracy of BPR and state of the art models. BPR achieves performance close to FiD (large) with almost half of the parameters.

didate generation (treating all passages as candidates) results in the same performance as using only top-1000 candidates, but significantly increases the query time due to the expensive inner product computation over all passage embeddings.

Comparison with State of the Art Table 3 presents the EM QA accuracy of BPR combined with state-of-the-art reader models. Here, we also report the results of our model using an improved reader based on ELECTRA-large (Clark et al., 2020) instead of BERT-base. Our improved model outperforms all models except the large model of Fusion-in-Decoder (FiD), which contains more than twice as many parameters as our model.

5 Conclusion

We introduce BPR, which is an extension of DPR, based on a learning-to-hash technique and a novel two-stage approach. It reduces the computational cost of open-domain QA without a loss in accuracy.

Acknowledgement

We are grateful for the feedback and suggestions from the anonymous reviewers and the members of the UW NLP group. This research was supported by Allen Distinguished investigator award, a gift from Facebook, and the Nakajima Foundation Fellowship.

References

Yue Cao, Bin Liu, Mingsheng Long, and Jianmin Wang. 2018. HashGAN: Deep Learning to Hash With Pair Conditional Wasserstein GAN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1287–1296.

Z Cao, M Long, J Wang, and P S Yu. 2017. HashNet: Deep Learning to Hash by Continuation. In *2017 IEEE International Conference on Computer Vision*, pages 5609–5618.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating Large-Scale Inference with Anisotropic Vector Quantization. In *International Conference on Machine Learning*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 3929–3938.

Gautier Izacard and Edouard Grave. 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. 2020. A Memory Efficient Baseline for Open Domain Question Answering. *arXiv preprint arXiv:2012.15156*.

H Jégou, M Douze, and C Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128.

J Johnson, M Douze, and H Jégou. 2019. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous Feature Learning and Hash Coding With Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and others. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33*.

Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. 2016. Feature Learning Based Deep Supervised Hashing with Pairwise Labels. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1711–1717.

Y A Malkov and D A Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Behnam Neyshabur and Nathan Srebro. 2015. On Symmetric and Asymmetric LSHs for Inner Product Search. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1926–1934.

J Wang, W Liu, S Kumar, and S Chang. 2016. Learning to Hash for Indexing Big Data—A Survey. *Proceedings of the IEEE*, 104(1):34–57.

J Wang, T Zhang, J Song, N Sebe, and H T Shen. 2018. A Survey on Learning to Hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):769–790.

Dong Xu and Wu-Jun Li. 2020. Hashing Based Answer Selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9330–9337.

Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1):2415–2421.

Appendix for "Efficient Passage Retrieval with Hashing for Open-domain Question Answering"

A Details of Experimental Setup

A.1 Knowledge Source

As the knowledge source, we use the preprocessed Wikipedia corpus consisting of 21,015,324 Wikipedia passages available on the website of Karpukhin et al. (2020). The corpus is based on the December 20, 2018 version of the English Wikipedia and created by filtering out semi-structured data (i.e., tables, infoboxes, lists, and disambiguation pages) and splitting the remaining Wikipedia articles into multiple, disjointed text blocks of 100 words each.

A.2 Question Answering Datasets

We conduct experiments using the NQ and TQA datasets with the training, development, and test sets as in Lee et al. (2019); Karpukhin et al. (2020). A brief description of these datasets is provided as follows:

- **NQ** is a QA dataset for which questions are obtained from Google queries and answers comprise the spans of English Wikipedia articles.
- **TQA** consists of trivia questions and their answers retrieved from the Web.

We use the preprocessed datasets available on the website of Karpukhin et al. (2020).[5] The numbers of questions contained in these datasets are listed in Table 4. For each question, the dataset contains three types of passages: (1) *positive passages* selected based on gold-standard human annotations or distant supervision, (2) *random negative passages* selected randomly from all the passages, and (3) *hard negative passages* selected based on the BM25 scores between the question and all the passages.

A.3 Details of BPR

Our training configuration follows that of Karpukhin et al. (2020). In particular, for each question, we use one positive and one hard negative passage and create a mini-batch comprising 128 questions. We use the method of *inbatch-negatives*, wherein each positive passage in a mini-batch is treated as the negative passage of each question in the mini-batch if it does not correspond to the question. Our model contains 220 million parameters, and is trained for up to 40 epochs using Adam. Regarding the hyperparameter search, we select the learning rate from the search range {1e-5, 2e-5, 3e-5, 5e-5} based on the top-100 recall on the validation set of the NQ dataset. Therefore, the number of hyperparameter search trials is 4. The detailed hyperparameters are listed in Table 5.

[5] https://github.com/facebookresearch/DPR

Dataset	Train	Validation	Test
NQ	58,880	8,757	3,610
TQA	60,413	8,837	11,313

Table 4: Number of questions in the preprocessed dataset used in our experiments.

Name	Value
Batch size	128
Maximum question length	256
Maximum passage length	256
Maximum training epochs	40
Peak learning rate	2e-5
Learning rate decay	linear
Warmup ratio	0.06
Dropout	0.1
Weight decay	0.0
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1e-6

Table 5: Hyperparameters used to train BPR.

A.4 Details of Reader

For each passage in the top-k passages retrieved by the retriever, the reader assigns a relevance score to the passage and selects the best answer span in the passage. The final answer is the selected span from the passage with the highest relevance score.

Let $\mathbf{P}_i \in \mathbb{R}^{q \times d}$ $(1 \leq i \leq k)$ be a BERT output representation for the i-th passage, where q is the maximum token length of the passage, and d is the dimension size of the output representation. The probabilities of a passage being selected and a token being the start or end positions of an answer is computed as

$$P_{score}(i) = \mathrm{softmax}\left(\hat{\mathbf{P}}^{\top}\mathbf{w}_{score}\right)_i, \quad (8)$$

$$P_{start,i}(s) = \mathrm{softmax}\left(\mathbf{P}_i\mathbf{w}_{start}\right)_s, \quad (9)$$

$$P_{end,i}(t) = \mathrm{softmax}\left(\mathbf{P}_i\mathbf{w}_{end}\right)_t, \quad (10)$$

where $\hat{\mathbf{P}} = [\mathbf{P}_1^{[\mathrm{CLS}]}, \ldots, \mathbf{P}_k^{[\mathrm{CLS}]}] \in \mathbb{R}^{d \times k}$, $\mathbf{w}_{score} \in \mathbb{R}^d$, $\mathbf{w}_{start} \in \mathbb{R}^d$, and $\mathbf{w}_{end} \in \mathbb{R}^d$.

Name	BERT-base	ELECTRA-large
Batch size	32	32
Maximum token length	350	350
Maximum training epochs	20	20
Negative passage size	23	17
Peak learning rate	2e-5	1e-5
Learning rate decay	linear	linear
Warmup ratio	0.06	0.06
Dropout	0.1	0.1
Weight decay	0.0	0.0
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Adam ϵ	1e-6	1e-6

Table 6: Hyperparameters used to train the reader based on BERT-base and that based on ELECTRA-large.

Configuration	Top 1	Top 20	Top 100
$\gamma = 0.025$	39.4	**76.7**	83.8
$\gamma = 0.05$	39.5	76.5	84.0
$\gamma = 0.1$	**39.8**	**76.7**	**84.1**
$\gamma = 0.2$	39.6	76.3	83.9

Table 7: Top-1, top-20, and top-100 recall of our model with $\gamma \in \{0.025, 0.05, 0.1, 0.2\}$ on the validation set of the NQ dataset.

The passage selection score of the i-th passage is given as $P_{score}(i)$, and the score of the s-th to t-th tokens from the i-th passage is given as $P_{start,i}(s) \times P_{end,i}(t)$.

During the training, we sample one positive and multiple negative passages from the passages returned by the retriever. The model is trained to maximize the log-likelihood of the correct answer span in the positive passage, combined with the log-likelihood of the positive passage being selected. We use the BERT-base or ELECTRA-large as our pretrained model. Regarding the hyperparameter search, we select the learning rate from {1e-5, 2e-5, 3e-5, 5e-5} based on its EM accuracy on the validation set of the NQ dataset. Therefore, the number of hyperparameter search trials is 4. Detailed hyperparameters are listed in Table 6.

B Effects of Scaling Parameter

To investigate how the scaling parameter, γ, affects the performance, we test the performance of our model using various γ values, where $\gamma \in \{0.025, 0.05, 0.1, 0.2\}$. The retrieval performance on the validation set of the NQ dataset is shown in Table 7. Overall, the scaling parameter has a minor impact on the performance. We select $\gamma = 0.1$ because of its enhanced performance.

#candidates	Top 1		Top 20		Top 100	
	NQ	TQA	NQ	TQA	NQ	TQA
$l = 200$	41.1	49.7	77.9	77.9	85.4	84.0
$l = 500$	41.1	49.7	77.9	77.9	85.6	84.4
$l = 1000$	41.1	49.7	77.9	77.9	**85.7**	**84.5**
$l = 2000$	41.1	49.7	77.9	77.9	**85.7**	**84.5**

Table 8: Top-1, top-20, and top-100 recall of our model with $l \in \{200, 500, 1000\}$ on test sets.

Configuration	Top 1	Top 20	Top 100
Cross entropy loss	28.6	67.8	79.8
Ranking loss $\alpha = 0.0$	39.8	76.4	84.0
Ranking loss $\alpha = 1.0$	40.0	76.5	84.0
Ranking loss $\alpha = 2.0$	39.8	**76.7**	**84.1**
Ranking loss $\alpha = 4.0$	**40.3**	**76.7**	84.0

Table 9: Top-1, top-20, and top-100 recall of our model with the various settings of the loss function $\mathcal{L}_{\text{cand}}$ evaluated on the validation set of the NQ dataset.

C Effects of Number of Candidate Passages

We report the performance of our model with the varied number of candidate passages l in Table 8. Overall, BPR achieves similar performance in all settings. Increasing the number of candidate passages slightly improves the top-100 performance until it reaches $l = 1000$.

D Effects of Loss of Task #1 with Various Settings

We investigate the effects of using various settings of the loss function $\mathcal{L}_{\text{cand}}$ in Eq.(6). Instead of using the ranking loss, we test the performance with the cross-entropy loss, similar to Eq.(2), and $\tilde{\mathbf{h}}_q$ and $\tilde{\mathbf{h}}_p$ are used instead of $\mathbf{e}_q$ and $\mathbf{e}_p$, respectively. Furthermore, we also test how the parameter α affects the performance. As shown in Table 9, the cross-entropy loss clearly performs worse than the ranking loss. Furthermore, a change in the parameter α has a minor impact on the performance. Here, we select the ranking loss with $\alpha = 2.0$ because of its enhanced performance on the top-20 and top-100 performance.

Entity Concept-enhanced Few-shot Relation Extraction

ShanYang[1], Yongfei Zhang[1,2,3]*, Guanglin Niu[1], Qinhua Zhao[4], Shiliang Pu[5]
[1]Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, BeiHang University, Beijing 100191, China
[2]State Key Laboratory of Virtual Reality Technology and Systems, BeiHang University,Beijing 100191, China
[3]Pengcheng Laboratory, Shenzhen 518055, China
[4]State Key Laboratory of Software Development Environment School of Computer Science and Engineering, BeiHang University, Beijing 100191, China
[5]Hikvision Research Institute, Hangzhou 311500, China
{shanyang, yfzhang, beihangngl, zhaoqh}@buaa.edu.cn, pushiliang.hri@hikvision.com

Abstract

Few-shot relation extraction (FSRE) is of great importance in long-tail distribution problem, especially in special domain with low-resource data. Most existing FSRE algorithms fail to accurately classify the relations merely based on the information of the sentences together with the recognized entity pairs, due to limited samples and lack of knowledge. To address this problem, in this paper, we proposed a novel entity CONCEPT-enhanced FEw-shot Relation Extraction scheme (ConceptFERE), which introduces the inherent concepts of entities to provide clues for relation prediction and boost the relations classification performance. Firstly, a concept-sentence attention module is developed to select the most appropriate concept from multiple concepts of each entity by calculating the semantic similarity between sentences and concepts. Secondly, a self-attention based fusion module is presented to bridge the gap of concept embedding and sentence embedding from different semantic spaces. Extensive experiments on the FSRE benchmark dataset FewRel have demonstrated the effectiveness and the superiority of the proposed ConceptFERE scheme as compared to the state-of-the-art baselines. Code is available at *https://github.com/LittleGuoKe/ConceptFERE*.

1 Introduction

Relation extraction (RE) is a fundamental task for knowledge graph construction and inference, which however often encounters challenges of long-tail distribution and low-resource data, especially in practical applications including medical or public security fields. In this case, it is difficult for existing RE models to learn effective classifiers (Zhang et al., 2019; Han et al., 2020). Therefore, FSRE has become a hot topic in both academia and industry. Existing FSRE methods can be roughly divided into two categories according to the type of adopted training data. The models of the first category only uses the plain text data, without any external information. The representative Siamese (Koch et al., 2015) and Prototypical (Snell et al., 2017) network in metric learning are used in the FSRE task to learn representation and metric function. BERT-PAIR (Gao et al., 2019b) pairs up all supporting instances with each query instance, and predicts whether the pairs are of the same category, which can be regarded as a variant of the Prototypical network. Gao (Gao et al., 2019a) and Ye (Ye and Ling, 2019) add the attention mechanism to enhance the prototype network. In order to alleviate the problem of insufficient training data, MICK (Geng et al., 2020) learns general language rules and grammatical knowledge from cross-domain datasets. Wang (Wang et al., 2020) proposes the CTEG model to solve the relation confusion problem of FSRE. Cong (Cong et al., 2020) proposes an inductive clustering based framework, DaFeC, to solve the problem of domain adaptation in FSRE.

Since the information of the plain text is limited in FSRE scenarios, the performance gain is marginal. Thus, the algorithms in the second category introduce external information, to compensate the limited information in FSRE, so as to enhance the performance. In order to improve the model's generalization ability for new relations, Qu (Qu et al., 2020) studies the relationship between different relations by establishing a global relation graph. The relations in the global relation graph

*Corresponding Author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 987–991
August 1–6, 2021. ©2021 Association for Computational Linguistics

Relation	founder
Sentence	**Microsoft** was founded by **Bill Gates** and Paul Allen on April 4, 1975
Head entity concept	company, vendor, client
Tail entity concept	person, billionaire, entrepreneur

Table 1: The bold words in the sentence correspond to the head entity and the tail entity.

come from Wikidata.[1] TD-Proto (Yang et al., 2020) introduces text descriptions of entities and relations from Wikidata to enhance the prototype network and provide clues for relation prediction.

Although the introduction of knowledge of text description can provide external information for FSRE and achieve state-of-the-art performance, TD-proto only introduces one text description in Wikidata for each entity. However, this might suffer from the mismatching between entity and text description and leads to the degraded performance. Besides, since the text description for each entity is often relatively long, it is not a easy job to extract the most useful information within the long text description.

In contrast to the long text descriptions, the concept is an intuitive and concise description of an entity and can be readily obtained from concept databases, like YAGO3, ConceptNet and Concept Graph, etc. Besides, the concept is more abstract than the specific text description for each entity, which is an idea compensation to the limited information in FSRE scenarios.

As shown in Table 1, intuitively knowing that the concept of head entity is a *company* and the concept of tail entity is an *entrepreneur*, the relation corresponding to the entity pair in the sentence can be limited to a range: *ceo*, *founder*, *inauguration*. On the other hand, some relations should be wiped out, e.g., *educated at*, *presynaptic connection*, *statement describes*. The semantic information of concept can assist determining the relation: *founder* predicted by the model.

To address the above challenges, we propose a novel entity CONCEPT-enhanced FEw-shot Relation Extraction scheme (ConceptFERE), which introduces the entity concept to provide effective clues for relation prediction. *Firstly*, as shown in Table 1, one entity might have more than one concept from different aspects or hierarchical levels and only one of the concepts might be valuable for final relation classification. Therefore, we design a concept-sentence attention module to choose the most suitable concept for each entity by comparing the semantic similarity of the sentence and each concept. *Secondly*, since the sentence embedding and pre-trained concept embedding are not learned in the same semantic space, we adopt the self-attention mechanism (Devlin et al., 2018) for word-level semantic fusion of the sentence and the selected concept for final relation classification. Experimental results on benchmark dataset show that our method achieves state-of-the-art FSRE performance.

[1] https://www.wikidata.org/

2 Model

2.1 System Overview

Figure 1 shows the structure of our proposed ConceptFERE. The sentence representation module uses BERT to obtain the sentence embedding, the concept representation adopts the pre-trained concept embedding (Shalaby et al., 2019), which uses the skip-gram model to learn the representation of the concept on the Wikipedia text and the Concept Graph. Relation classifier can be implemented by the fully connected layer. The remaining modules of the model will be described in detail below.

2.2 Concept-Sentence Attention Module

Intuitively, one needs to pay more attention to the concept of high semantic correlation with the sentence, which can provide more effective clues for RE. Firstly, since the pre-trained concept embedding (v_c) and sentence embedding (v_s) are not learned in the same semantic space, we can not compare the semantic similarity directly. So the semantic transformation is performed by multiplying the v_c and v_s by the projection matrix P to get their representations v_cP and v_sP in the same semantic space, where P can be learned by fully connected networks. Secondly, by calculating the semantic similarity between sentence and each concept of entity, the similarity value is obtained the dot product of the concept embedding v_c and the sentence embedding v_s as similarity sim_{cs}. Finally, in order to select a suitable concept from the calculated similarity value, we design the 01-GATE.

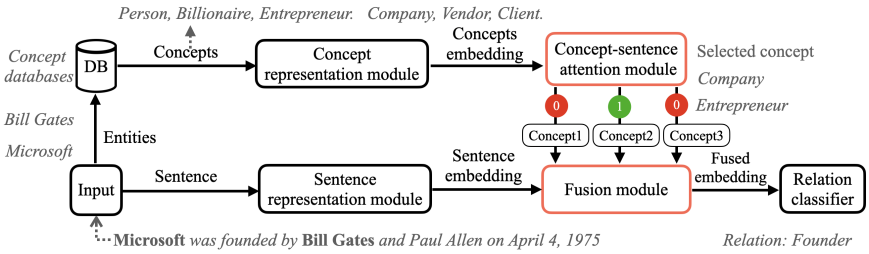


Figure 1: Structure diagram of ConceptFERE model.

The similarity value is normalized by the Softmax function. If sim_{cs} is less than the set threshold α, 01-GATE assigns 0 to the attention score of the corresponding concept, and this concept will be excluded in subsequent relation classification. We choose the suitable concept with the attention score of 1, which is used as a effective clue to participate in relation prediction.

2.3 Self-Attention based Fusion Module

Since concept embedding and the embedding of words in sentences are not learned in the same semantic space, we design a self-attention (Devlin et al., 2018) based fusion module to perform word-level semantic fusion of the concept and each word in the sentence. First, the embedding of all words in the sentence and the selected concept embedding are concatenated, and then fed to the self-attention module. As shown in Figure 2, the self-attention module calculates the similarity value between the concept and each word in the sentence. It multiplies the concept embedding and the similarity value, and then combine with its corresponding word embedding as follow:

$$fusion_{v_i} = \sum_{j=1}^{N} sim\left(q_i, k_j\right) v_j \qquad (1)$$

where $fusion_{v_i}$ represents the embedding of v_i after v_i performs the word-level semantic fusion. The q_i, k_j, and v_j are derived from self-attention, they represent the concept embedding or the word embedding.

3 Experiment

3.1 Dataset, Evaluation and Comparable Models

Dataset: In order to verify our proposed method, we use the most commonly used FSRE dataset FewRel (Han et al., 2018), which contains 100 relations and 70,000 instances extracted from

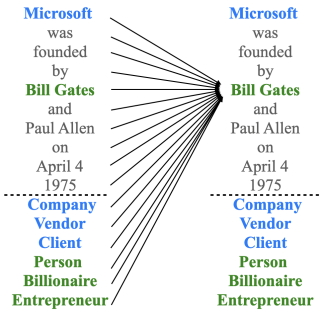


Figure 2: The word-level semantic fusion.

Wikipedia, with 20 relations in the unpublished test set. So we follow previous work (Yang et al., 2020) to re-split the published 80 relations into 50, 14 and 16 for training, validation and testing, respectively.

Evaluation: N-way-K-shot (N-w-K-s) is commonly used to simulate the distribution of FewRel in different situations, where N and K denote the number of classes and samples from each class, respectively. In N-w-K-s scenario, accuracy is used as the performance metric.

Comparable Models: We choose excellent baseline models, GNN (Garcia and Bruna, 2017), SNAIL (Mishra et al., 2017), Proto (Snell et al., 2017), HATT-Proto (Gao et al., 2019a), MLMAN (Ye and Ling, 2019) and TD-proto (Yang et al., 2020) for comparison, and their experimental results are derived from (Yang et al., 2020).

3.2 Model Training Details

The BERT parameters are initialized by bert-base-uncased, and the hidden size is 768. The threshold α is 0.7. Hyperparameters such as learning rate follow the settings in (Gao et al., 2019b). The entity concept is obtained from Concept Graph[2]. Concept Graph is a large-scale common sense conceptual knowledge graph developed by Microsoft, which contains concept of entities stored in triplets (Entity,

[2]https://concept.research.microsoft.com/Home/Download

Model	Encoder	5-w-1-s	5-w-5-s	10-w-1-s	10-w-5-s
GNN (Garcia and Bruna, 2017)	CNN	67.30	78.84	54.10	62.89
SNAIL (Mishra et al., 2017)	CNN	71.13	80.04	50.61	66.68
Proto (Snell et al., 2017)	CNN	74.29	85.18	61.15	74.41
HATT-Proto (Gao et al., 2019a)	CNN	74.84	85.81	62.05	75.25
MLMAN (Ye and Ling, 2019)	CNN	78.21	88.01	65.70	78.35
Bert-PAIR (Gao et al., 2019b)	BERT	82.57	88.47	73.37	81.10
TD-Proto (Yang et al., 2020)	BERT	84.76	92.38	74.32	85.92
ConceptFERE	BERT	**89.21**	–	**75.72**	–
ConceptFERE (Simple)	BERT	84.28	90.34	74.00	81.82

Table 2: Accuracies (%) of different models on test set.

IsA, Concept) and can provide concept knowledge for entities in ConceptFERE. The concept embedding adopts the pre-trained concept embedding[3] (Shalaby et al., 2019).

Our proposed scheme is implemented on top of BERT-PAIR, since the concept provided by ConceptFERE can be used as an effective clue.

3.3 Performance and Comparisons

Table 2 tabulates the performance of different comparable models on the test set, where the algorithms in the first group are those state-of-the-art schemes without using any external information, while the TD-Proto in the second group uses external information of text descriptions of entities, and finally our proposed scheme in the third group. It should be noted that, due to the insufficient computing power of our GPU, the performance of the proposed ConceptFERE scheme is tested only under 5way1shot and 10way1shot scenarios. It can be observed from Table 2 that the proposed ConceptFERE model achieves the best performance, as compared to all the comparable schemes. More specifically, ConceptFERE achieves respectively 4.45 and 1.4 gains over the latest TD-Proto using external entity descriptions. And a performance gain of 6.64 and 2.35 is registered as compared to Bert-PAIR, the best model in the first category, under the 5way1shot and 10way1shot scenarios, respectively. This might due to that the generalization ability of concepts is stronger than text description and it is more suitable for FSRE. In theory, 1-shot relation extraction is a more difficult task than 5-shot relation extraction. The experimental results of 1-shot relation extraction have illustrated the effectiveness and superiority of our approach. We believe that our ConceptFERE scheme would also achieve the best performance under the other two scenarios.

[3]https://sites.google.com/site/conceptembeddings/

Model	5-w-1-s
ConceptFERE (bert)	89.21
w/o FUSION	83.11
w/o ATT	84.03
w/o ATT & FUSION	82.57

Table 3: Results of ablation study with ConceptFERE.

3.4 Ablation Study

In this section, to verify the effectiveness of the proposed concept-sentence attention module and self-attention based fusion module, presented in 2.2 and 2.3, respectively. As shown in Table 3, without using the concept-sentence attention and fusion module, the model performance of ConceptFERE (simple) drops sharply. This proves that the proposed concept-sentence attention module (ATT) and fusion module (FUSION) can effectively select appropriate concepts and perform word-level semantic integration of concepts and sentences. On the other hand, we present a simplified version of the ConceptFERE model, denoted as ConceptFERE (Simple), in which both the concept selection and fusion module are removed and the concepts and sentences are concatenated and inputted into the relation classification model. Specifically, we can input the concatenated sentences and concepts into BERT-PAIR (Gao et al., 2019b). As shown in Table 2, ConceptFERE (simple) achieves much better performance as compared to Bert-PAIR, the best model in the first category, under all four scenarios. This further validates the effectiveness of introducing the concept in enhancing the RE performance. More importantly, it can be easily applied to other models. As mentioned above, we only need to in-

put the concatenated entity concepts and sentences into the model.

4 Conclusion

In this paper, we have studied the FSRE task and presented a novel entity concept-enhanced FSRE scheme (ConceptFERE). The concept-sentence attention module was designed to select the appropriate concept from multiple concepts corresponding to each entity, and the fusion module was designed to integrate the concept and sentence semantically at the word-level. The experimental results have demonstrated the effectiveness of our method against state-of-the-art algorithms. As a future work, the commonsense knowledge of the concepts as well as the possible relations between them will be explicitly considered to further enhance the FSRE performance.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61772054, 62072022), and the NSFC Key Project (No. 61632001) and the Fundamental Research Funds for the Central Universities.

References

Xin Cong, Bowen Yu, Tingwen Liu, Shiyao Cui, Hengzhu Tang, and Bin Wang. 2020. Inductive unsupervised domain adaptation for few-shot classification via clustering. *arXiv preprint arXiv:2006.12816*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019a. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019b. Fewrel 2.0: Towards more challenging few-shot relation classification. *arXiv preprint arXiv:1910.07124*.

Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*.

Xiaoqing Geng, Xiwen Chen, Kenny Q Zhu, Libin Shen, and Yinggong Zhao. 2020. Mick: A meta-learning framework for few-shot relation classification with small training data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 415–424.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. *arXiv preprint arXiv:2004.03186*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *International Conference on Machine Learning*, pages 7867–7876. PMLR.

Walid Shalaby, Wlodek Zadrozny, and Hongxia Jin. 2019. Beyond word embeddings: learning entity and concept representations from large scale knowledge bases. *Information Retrieval Journal*, 22(6):525–542.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087.

Yingyao Wang, Junwei Bao, Guangyi Liu, Youzheng Wu, Xiaodong He, Bowen Zhou, and Tiejun Zhao. 2020. Learning to decouple relations: Few-shot relation classification with entity-guided attention and confusion-aware training. *arXiv preprint arXiv:2010.10894*.

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 2020. Enhance prototypical network with text descriptions for few-shot relation classification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2273–2276.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. *arXiv preprint arXiv:1906.06678*.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. *arXiv preprint arXiv:1903.01306*.

Improving Model Generalization: A Chinese Named Entity Recognition Case Study

Guanqing Liang, Cane Wing-Ki Leung
Wisers AI Lab, Wisers Information Limited
{quincyliang, caneleung}@wisers.com

Abstract

Generalization is an important ability that helps to ensure that a machine learning model can perform well on unseen data. In this paper, we study the effect of data bias on model generalization, using Chinese Named Entity Recognition (NER) as a case study. Specifically, we analyzed five benchmarking datasets for Chinese NER, and observed the following two types of data bias that may compromise model generalization ability. Firstly, the test sets of all the five datasets contain a significant proportion of entities that have been seen in the training sets. These test data are therefore not suitable for evaluating how well a model can handle unseen data. Secondly, all datasets are dominated by a few fat-head entities, i.e., entities appearing with particularly high frequency. As a result, a model might be able to produce high prediction accuracy simply by keyword memorization. To address these data biases, we first refine each test set by excluding seen entities from it, so as to better evaluate a model's generalization ability. Then, we propose a simple yet effective entity rebalancing method to make entities within the same category distributed equally, encouraging a model to leverage both name and context knowledge in the training process. Experimental results demonstrate that the proposed entity resampling method significantly improves a model's ability in detecting unseen entities, especially for company, organization and position categories.

1 Introduction

Named Entity Recognition (NER) is a fundamental building block for various downstream natural language processing tasks such as relation extraction (Bunescu and Mooney, 2005), event extraction (Ji and Grishman, 2008), information retrieval (Chen et al., 2015), question answering (Diefenbach et al., 2018), etc. Due to the ambiguous word boundaries and complex composition (Gui et al., 2019), Chinese NER task is more challenging compared with English NER.

Recently, by leveraging upon the pretrained language model (e.g, BERT (Devlin et al., 2018), etc.), we have witnessed superior performances on Chinese NER datasets, including: MSRA, Weibo, Ontonotes 4.0 and Resume (Li et al., 2020, 2019; Xuan et al., 2020). Despite the superior performance of the fine-tuned models, we argue that there are two types of data bias that can compromise the model generalization ability.

First, we observe that in widely used Chinese NER datasets, 50% to 70% entities in test data are seen in the training data. Such test data would therefore not be able to evaluate the true generalization ability of a model.

Second, the datasets are dominated by a few fat-head entities, i.e., entities appearing with particularly high frequency. For example, within the organization category of Cluener (Xu et al., 2020), fat-head entity 曼联 (Manchester United) appears 59 times, while 法兰克福队 (Eintracht Frankfurt) occurs only once. As a result, a model might be encouraged to memorize those fat-head entities rather than leveraging context knowledge during training process. The rationale is that given the same entity and diverse contexts, the easiest way for model convergence is to memorize the entity rather than extracting patterns from the diverse contexts.

To address these data biases, we first refine each test set by excluding seen entities from it, so as to better evaluate a model's generalization ability. Then, we propose a simple yet effective entity rebalancing method to make entities within the same category distributed equally, encouraging a model to leverage both name and context knowledge in the training process.

The contributions of this paper are as follows.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 992–997
August 1–6, 2021. ©2021 Association for Computational Linguistics

Dataset	Categories	Train	Dev	Test
MSRA	LOC, ORG, PER	41728	4636	4365
OntoNotes 4.0	GPE, LOC, ORG, PER	15724	4301	4346
Resume	CONT, EDU, LOC, NAME, ORG, PRO, RACE, TITLE	3821	463	477
Weibo	GPE.NAM, GPE.NOM, LOC.NAM, LOC.NOM, ORG.NAM, ORG.NOM, PER.NAM, PER.NOM	1350	270	270
Cluener	movie, organization, company, game, book, scene, name, government, address, position	10748	1343	1345

Table 1: Chinese NER datasets overview: entity categories and the sentence number in train/dev/test data.

- We analyze five benchmarking Chinese NER datasets and identify two types of data bias that can compromise model generalization ability.

- We refine each test set by excluding seen entities from it, which can measure real model generalization. Specifically, the competitive BERT+CRF model only achieves 33.33% and 65.10% F1 score on detecting unseen organization entities of Cluener and MSRA dataset respectively, which are far from satisfactory.

- We design a simple yet effective algorithm to rebalance the entity distribution. The experiments show that the proposed method significantly improves the model generalization. In particular, the F1 score has been improved by 12.61% and 37.14% on the organization category of Cluener and MSRA dataset respectively.

2 Dataset Observation

2.1 Dataset Overview

In this study, we analyze five benchmarking Chinese NER datasets, including: (1) MSRA (Levow, 2006), (2) Ontonotes 4.0 (Weischedel et al.), (3) Resume (Zhang and Yang, 2018), (4) Weibo (Peng and Dredze, 2015) and (5) Cluener (Xu et al., 2020). The statistics of these datasets are shown in Table 1.

2.2 Seen vs Unseen Entity

If an entity in dev/test data has been covered by the training data, we refer it as a seen entity. Otherwise, it is an unseen entity. To quantify the degree to which entities in the dev/test data have been seen in the training data, we define a measurement called entity coverage ratio. The entity coverage ratio of data D^{te} is denoted by $r(D^{te})$, which is calculated using the below equation.

$$r(D^{te}) = \frac{|Ent(D^{te}) \bigcap Ent(D^{train})|}{|Ent(D^{train})|} \quad (1)$$

where $Ent(.)$ denotes a function to obtain the list of annotated entities and D^{train} represents the training data. As Table 2 shows, the entity coverage ratios of the dev and test data in different benchmarking datasets are very high, ranging from 0.429 to 0.709.

Dataset	r(dev)	r(test)
MSRA	0.554	0.709
OntoNotes 4.0	0.505	0.514
Resume	0.540	0.544
Weibo	0.498	0.429
Cluener	0.615	-

Table 2: Entity coverage ratio of dev and test data in different Chinese NER datasets.

Observation 1 The test sets of Chinese NER datasets contain a significant proportion of seen entities.

2.3 Fat-head vs Long-tail Entity

Fat-head entity is defined as the entity appearing with particularly high frequency, while long-tail entity is defined as the entity with very few mentions. To identify the existence of fat-head entity, we use kurtosis (Balanda and MacGillivray, 1988), a statistical measure that defines how heavily the tails of a distribution differ from the tails of a normal distribution. Usually, high kurtosis (greater than 3) indicates the existence of outliers, i.e., fat-head entities.

Table 3 shows the kurtosis score of each category in different datasets. For example, the kurtosis score of PER category of training data in MSRA dataset is 984.1, which is very high. We find that 1% distinct entities with the highest frequency contribute 21% of the overall annotation.

Observation 2 Fat-head entities prevail in different categories of Chinese NER datasets.

We think this finding is also valid in other NER datasets, since the annotated corpus is usually collected within a certain time frame when some entities (e.g., celebrities, organizations) get much more exposure than others.

We hypothesize that the dominance of fat-head entities will cause the model to simply memorize

those high-frequency entities without fully leveraging context knowledge. The rationale is that given the same entity and diverse contexts, the easiest way for model convergence is to memorize the entity rather than extracting patterns from the diverse contexts.

Dataset	Training	Test
MSRA	ORG : 609.2 PER : 984.1 LOC : 1272.1	ORG : 66.3 LOC : 226.6 PER : 387.4
OntoNotes 4.0	LOC : 46.9 PER : 77.2 GPE : 108.9 ORG : 151.4	LOC : 36.9 PER : 79.0 GPE : 184.7 ORG : 337.5
Resume	LOC : 0.0 RACE : 4.2 EDU : 11.1 CONT : 11.8 PRO : 45.6 NAME : 59.6 TITLE : 239.3 ORG : 1723.1	CONT : 1.0 RACE : 1.0 LOC : 3.3 EDU : 4.5 PRO : 9.1 NAME : 35.6 TITLE : 53.5 ORG : 212.4
Weibo	GPE.NOM : 1.5 LOC.NAM : 6.4 LOC.NOM : 8.0 ORG.NOM : 9.9 GPE.NAM : 13.4 PER.NOM : 38.5 ORG.NAM : 101.1 PER.NAM : 188.4	GPE.NOM : 0.0 ORG.NOM : 1.7 GPE.NAM : 4.5 ORG.NAM : 5.0 LOC.NOM : 6.1 LOC.NAM : 6.6 PER.NOM : 27.7 PER.NAM : 48.0
Cluener	movie : 25.4 organization : 35.3 company : 81.4 game : 90.2 book : 97.5 scene : 117.4 name : 261.8 government : 308.2 address : 511.0 position : 570.0	-

Table 3: Kurtosis score of different categories in various Chinese NER datasets.

3 Method

To improve model's generalization ability in detecting unseen entities, we argue that the model should be trained to leverage both name and context knowledge (Nie et al., 2020; Lin et al., 2020). Thus, we propose a simple yet effective entity rebalancing algorithm. The main idea is to make the annotated entity equally distributed within the same category.

There are two major reasons why the proposed entity rebalancing algorithm works. First, the equal distribution will encourage the model to leverage both name knowledge and context knowledge, since there are no simple statistical cues (Niven and Kao, 2019) to exploit due to uneven distribution. Second, different entities within the same category should be interchangeable semantically in most cases, which avoids the train-test discrepancy.

The proposed algorithm works as follows. First, rebalance the annotated entity frequency in the training data. Let C_l denotes the original entity frequency counter of category l. For example, given $C_l = \{e_1 : 11, e_2 : 1, e_3 : 1\}$, which means entity e_1 is annotated 11 times, and both e_2 and e_3 are annotated once in the category l, which is very imbalanced. Then we turn C_l to the balanced entity frequency counter C_l^b, which is $C_l^b = \{e_1 : 5, e_2 : 4, e_3 : 4\}$. In C_l^b, the difference between the maximum and minimum entity frequency is 1 at most. Second, replace the fat-head entity with randomly sampled entity of the same category, once its accumulated occurrence surpasses the rebalanced frequency in C_l^b. Details are shown in **Algorithm 1**.

Algorithm 1: Entity replacement algorithm

```
foreach sentence in Dataset do
    foreach ent_text, ent_label in sentence do
        l = ent_label;
        if C_l^b[ent_text] > 0 then
            keep ent_text as it is;
            C_l^b[ent_text] -= 1;
        else
            sample ent_s from c_l^b if c_l^b[ent_s] > 0;
            replace ent_text with ent_s;
            C_l^b[ent_s] -= 1;
```

4 Experiments

4.1 Experiment Settings

According to observation 1, the test sets of Chinese NER datasets contain a significant proportion of seen entities, which fails to evaluate the true model generalization ability. In our study, the test sample will be excluded if it contains entities that are covered in training data. For Cluener (Xu et al., 2020), we split the original training set into 90% train and 10% dev, and use the development set for test, as the test set is not publicly available. For Resume (Zhang and Yang, 2018) and Weibo (Peng and Dredze, 2015) datasets, we report evaluation results on the selected categories, since there are zero or very few unseen entities on other categories.

We use the BIOES tagging scheme to label named entities, since previous studies have shown optimistic improvement with this scheme (Ratinov and Roth, 2009). We report span-level micro-averaged F1 score obtained from `seqeval` (Nakayama, 2018) toolkit using IOBES scheme.

We use BERT+CRF as the model architecture. In particular, we use bert-base-chinese pre-

trained model [1] (12-layer, 768-hidden, 12-heads) released by google (Devlin et al., 2018). The hyperparameters of the model are tuned on the development set using grid search method (details are reported in Appendix A.). As shown in Table 4, the adopted BERT+CRF model is competitive with the complicated state-of-the-art models.

Model	MSRA	OntoNotes	Resume	Weibo
Glyce-BERT (Meng et al., 2019)	95.54	81.63	96.54	67.60
BERT+FLAT (Li et al., 2020)	96.09	81.82	95.86	68.55
BERT+CRF (Ours)	95.57	**82.29**	95.71	**69.89**

Table 4: Comparison between BERT+CRF and the state-of-the-art models using the same train/dev/test splits as (Li et al., 2020, 2019; Xuan et al., 2020)

4.2 Results

Table 5 presents the comparisons between the proposed method and the baseline on five Chinese NER datasets. The baseline uses the original training data, while the proposed applies entity rebalancing algorithm on the original training data.

For Cluener, MSRA and OntoNotes datasets with over 10K training samples, our proposed method outperforms the baseline on different categories. One exception is on the address category of Cluener dataset when the proposed method performs worse than the baseline by -2.58%. We believe it is due to the fact that the address category contains both geopolitical entities and location entities, which are not interchangeable semantically.

For Weibo dataset, the proposed outperforms the baseline by 8.89% in PER.NAM category, but performs worse in PER.NOM category. Note that the PER.NOM category contains entities such as man, woman and friend, which are hard to generalize based on context knowledge. For Resume dataset, the proposed method does not work well. We think it is due to the structure of the resume corpus, which is the mere concatenation of name, education and organization, etc. Thus, there is very few context knowledge to leverage.

Overall, the proposed entity rebalancing method is able to improve model's generalization ability in detecting unseen entities. However, the proposed method only works for categories which meet certain conditions. First, the entities of the same category require to be interchangeable semantically. Second, the entities should be dependent of context knowledge.

Cluener			
Category	**Baseline**	**Proposed**	**F1 Improvement**
address	58.48	56.97	-2.58%
book	77.65	83.72	+7.82%
company	62.34	64.86	+4.04%
game	61.29	62.50	+1.97%
government	80.00	83.78	+4.72%
movie	71.91	75.61	+5.15%
name	74.38	75.81	+1.92%
organization	33.33	45.71	+37.14%
position	35.90	52.63	+46.60%
scene	74.56	78.31	+5.03%
MSRA			
Category	**Baseline**	**Proposed**	**F1 Improvement**
LOC	86.79	89.17	+2.74%
ORG	89.69	89.69	+0.00%
PER	95.85	96.35	+0.52%
OntoNotes 4.0			
Category	**Baseline**	**Proposed**	**F1 Improvement**
GPE	64.93	66.94	+3.10%
LOC	37.88	45.03	+18.88%
ORG	65.10	73.31	+12.61%
PER	96.45	96.32	-0.13%
Weibo			
Category	**Baseline**	**Proposed**	**F1 Improvement**
PER.NAM	69.09	75.23	+8.89%
PER.NOM	46.67	45.28	-2.98%
Resume			
Category	**Baseline**	**Proposed**	**F1 Improvement**
NAME	1.00	1.00	0%
ORG	90.62	87.88	-3.02%

Table 5: Evaluation results (F1 score) of the proposed entity resampling method and the baseline on unseen test data

5 Conclusion and Future Work

In this paper, we take Chinese NER as a case study, aiming to improve the model generalization by mitigating the data bias. We first refine each test set by excluding seen entities from it, so as to better evaluate a model's generalization ability. Then, we propose an entity rebalancing method to make entities within the same category distributed equally. Experimental results show that the proposed entity rebalancing method significantly improves a model's ability in detecting unseen entities.

As future work, we will first investigate the generalizability of this study to non-Chinese NER. Second, we will improve the entity replacement algorithm by leveraging language model so that the replaced entity is more semantically plausible.

[1] https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip.

References

K. Balanda and H. MacGillivray. 1988. Kurtosis: A critical review. *The American Statistician*, 42:111–119.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. 2018. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.*, 55(3):529–569.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuan-Jing Huang. 2019. A lexicon-based graph neural network for chinese ner. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1039–1049.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.

Hongyu Lin, Yaojie Lu, Jialong Tang, Xianpei Han, Le Sun, Zhicheng Wei, and Nicholas Jing Yuan. 2020. A rigorous study on named entity recognition: Can fine-tuning pretrained model lead to the promised land? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7291–7300, Online. Association for Computational Linguistics.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems*, volume 32, pages 2746–2757. Curran Associates, Inc.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Yuyang Nie, Yuanhe Tian, Xiang Wan, Yan Song, and Bo Dai. 2020. Named entity recognition for social media texts with semantic augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554, Lisbon, Portugal. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, et al. Ontonotes release 4.0.

Liang Xu, Qianqian Dong, Cong Yu, Yin Tian, Weitang Liu, Lu Li, and Xuanwei Zhang. 2020. Cluener2020: Fine-grained name entity recognition for chinese. *arXiv preprint arXiv:2001.04351*.

Zhenyu Xuan, Rui Bao, Chuyu Ma, and Shengyi Jiang. 2020. Fgn: Fusion glyph network for chinese named entity recognition. *arXiv preprint arXiv:2001.05272*.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

Appendix A: Hyper-parameter Settings

Parameter	Value
learning_rate	[5e-5, 3e-5, 2e-5]
warmup_proportion	[0]
train_batch_size	[32]
seed	[2020, 1, 10]
crf_learning_rate	[1e-3, 1e-4]
model_name_or_path	["bert-base-chinese"]
max_seq_length	[128]
eval_batch_size	[16]
num_train_epochs	[10]
weight_decay	[0]
is_learning_rate_linearly_decrease	["yes"]

Table 6: The range of hyper-parameters grid-search for BERT+CRF model.

Three Sentences Are All You Need: Local Path Enhanced Document Relation Extraction

Quzhe Huang, Shengqi Zhu, Yansong Feng*, Yuan Ye, Yuxuan Lai, Dongyan Zhao
Wangxuan Institute of Computer Technology, Peking University, China
The MOE Key Laboratory of Computational Linguistics, Peking University, China
{huangquzhe, zhusq, fengyansong, pkuyeyuan, erutan, zhaody}
@pku.edu.cn

Abstract

Document-level Relation Extraction (RE) is a more challenging task than sentence RE as it often requires reasoning over multiple sentences. Yet, human annotators usually use a small number of sentences to identify the relationship between a given entity pair. In this paper, we present an embarrassingly simple but effective method to heuristically select evidence sentences for document-level RE, which can be easily combined with BiLSTM to achieve good performance on benchmark datasets, even better than fancy graph neural network based methods. We have released our code at https://github.com/AndrewZhe/Three-Sentences-Are-All-You-Need.

1 Introduction

The task of relation extraction (RE) focuses on extracting relations between entity pairs in texts, and has played an important role in information extraction. While earlier works focus on extracting relations within a sentence (Lin et al., 2016; Zhang et al., 2018), recent studies begin to explore RE at document level (Peng et al., 2017; Zeng et al., 2020a; Nan et al., 2020a), which is more challenging as it often requires reasoning across multiple sentences.

Compared with sentence level extraction, documents are significantly longer with useful information scattered in a larger scale. However, given a pair of entities, one may only need a few sentences, not the entire document, to infer their relationship; reading the whole document may not be necessary, since it may introduce unrelated information inevitably. As we can see in Figure 1, $S[1]$ is sufficient to recognize *Finland* as the country of *Espoo*, and recognizing the rest two instances requires just 2 sentences as supporting evidence as well. Although the document contains 6 sentences and evidence may span from $S[1] \sim S[6]$, identifying *each* relation instance can be achieved by just reading through 1 or 2 related sentences. This naturally leads us to consider a question: *given an entity pair, how many sentences are required to identify a relationship between them?* We perform a pilot study across 3 widely-used document RE datasets, DocRED (Yao et al., 2019), CDR (Li et al., 2016) and GDA (Wu et al., 2019). As shown in Table 1, we find that more than 95% instances require no more than 3 sentences as supporting evidence, and 87% even requires only 2 or less.

Espoo Cathedral
[1] *The Espoo Cathedral* is a medieval stone church in *Espoo*, *Finland* and the seat of the Diocese of Espoo of the Evangelical Lutheran Church of Finland. **[2]** The cathedral is located in the district of *Espoon keskus*, near the Espoonjoki river. … **[6]** In addition to being the seat of the Diocese of Espoo, *the Espoo Cathedral* serves as the church for *the EC Parish* and hosts …

Subject:	*Espoo*	*The Espoo Cathedral*	*the EC Parish*
Object:	*Finland*	*Espoon keskus*	*Finland*
Relation:	**country**	**location**	**country**
Evidence:	**[1]**	**[1], [2]**	**[1], [6]**

Figure 1: A case extracted from the DocRED dataset. While the document has 6 sentences, only 1 or 2 sentences form the evidence for each relation instance.

Our preliminary finding suggests that, instead of taking the entire document as context, a case-specific selection may be more useful to help a model focus on the most relevant and informative evidence. Previous studies apply graph neural networks (GNNs) for this filtering process (Christopoulou et al., 2019; Zeng et al., 2020b). Here, GNNs are used to collect relevant information from the entire context through an aggregation scheme (Nan et al., 2020a) and achieve great performance, but the selection of crucial evidence from documents is still implicit and lacks interpretability. If, as indicated by our pilot study, most entity relationships can be decided with just $1 \sim 3$ evidence sentences, is there a simpler method that can filter the document explicitly while maintaining the

* Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 998–1004
August 1–6, 2021. ©2021 Association for Computational Linguistics

	0	1	2	3	>=4	# Sent
DocRED	3.7%	49.7%	34.3%	8.4%	3.8%	8.0
CDR	0.0%	68.0%	30.0%	0.0%	2.0%	9.7
GDA	0.0%	66.0%	19.0%	3.0%	5.0%	10.2

Table 1: The proportion of instances with different supporting evidence sizes. # Sent shows the average number of sentences in a document.

crucial information?

We take a closer look at how entity pairs are contextually related in the annotated supporting evidence, and find that annotators tend to select sentences that can connect the two entities. We therefore design three heuristic rules to extract a small set of *paths* from the document, which can be seen as an approximation of the supporting evidence. Specifically, the *Consecutive Paths* consider the scenario where the head and tail entities are close in the context: if they are within 3 consecutive sentences, we regard these sentences as one path. The *Multi-Hop Paths* correspond to the entity pairs in distant sentences, which can be bridged via other entities that co-occur with the head entity and tail entity in different sentences. As the third relation in Figure 1 shows, *Finland* co-occurs with *The Espoo Cathedral* in S[1] and with *the EC Parish* in S[6], which makes it a bridge to connect *The Espoo Cathedral* and *the EC Parish*. In this case, S[1] and S[6] compose a multi-hop path. When neither of the above rules applies, we collect all the pairs of sentences where one contains head entity and the other contains tail entity as *Default Paths*.

By comparing our path set with human-annotated supporting evidence, we find that up to 87.5% of the supporting evidence can be fully covered by our heuristically selected paths. In other words, our straightforward and interpretable rules serve as an effective proxy to select supporting evidence from documents. We further feed our selected paths to a simple neural network model and obtain surprisingly good performance on DocRED, showing that our selected evidence can retain sufficient information from the entire document to support document-level relation extraction.

2 Do we need the entire document?

For document RE, the major challenge is that the subject and object involved in a relationship may appear in different sentences. Thus, more than one sentence is required to capture the relations. Nonetheless, how many sentences from the entire document are required to identify the relationship between an entity pair? To address this question, we analyze the supporting evidence presented in DocRED. The supporting evidence for a relation instance refers to all the sentences that can be used to decide whether this relation holds between the entity pair, labeled by human annotators (Yao et al., 2019). Table 1 shows the proportions of entity relation instances with different number of supporting sentences. As can be seen, more than 96% of the DocRED instances are associated with at most 3 supporting evidence. These only take up 37.5% of a document, since the average document length is 8 sentences. This means that reading a small part of a document is adequate for one to identify an entity relation instance.

We further extend our study to two widely used document RE datasets, CDR (Li et al., 2016) and GDA (Wu et al., 2019), where CDR is manually constructed and GDA is distantly supervised. In order to find the minimal number of sentences required, we ask annotators to label a minimal set of sentences that are exactly sufficient to identify an entity relation instance, instead of including all relation-associated sentences as the original DocRED pattern. We randomly select 100 instances respectively from CDR and GDA for this further annotation, and the results are shown at the bottom of Table 1[1]. Although the average length of documents in GDA and CDR are longer than DocRED, it turns out that one can still use no more than 3 supporting sentences to identify over 95% of the entity relation instances. The results on CDR and GDA confirm our previous finding that, a very small number of sentences (or more exactly, no more than 3 sentences) would make it sufficient for human annotators to recognize almost all entity relation instances in a document in widely-used benchmark datasets.

3 Which sentences are decisive?

Now our question is how to select the supporting sentences that are sufficient to identify an entity relation instance. Intuitively, the supporting evidence should be the sentences that build up the *connection* between a pair of entities. Thus, we aim to extract sentence *paths* from the head entity to the tail entity to describe how they are connected. As for the simplest case, if there exists one sentence that contains

[1] As GDA is a distantly supervised dataset, 7 instances that are found wrongly labeled are discarded.

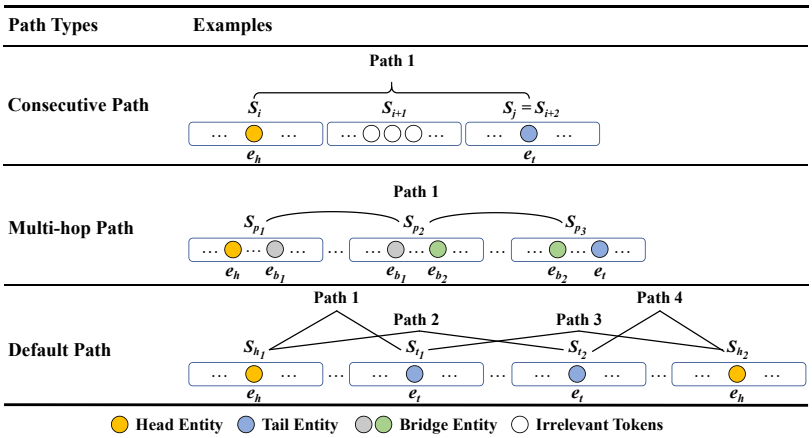


Figure 2: Types of paths connecting head and tail entities. The rounded rectangles represent sentences and the circles are mentions of involved entities or other irrelevant tokens. e_h and e_t stands for a mention of head and tail entities respectively, and S_* represents a sentence.

both the head and tail entities, the sentence itself can be seen as a path (the intra-sentence case). For more complex situations where the head and tail entities do not co-occur in one sentence, we define the following 3 types of paths which indicate how the head and tail entities can be possibly related in the context. Figure 2 provides a visualization of the three types of paths.

Consecutive Paths Previous studies have shown that the majority of inter-sentence relations are often in nearby text (Swampillai and Stevenson, 2010; Quirk and Poon, 2017). We thus select the consecutive sentences to form a path when the head and tail entities are in nearby sentences. Formally, if one mention of the head entity appears in sentence S_i and one mention of the tail entity is in sentence S_j, these two sentences along with the sentence in between, i.e., sentence S_{i+1}, ..., S_{j-1} (or S_{j+1}, ..., S_{i-1} when $i \geq j$) forms a possible path that connects the two entities.

Given that no more than 3 sentences would suffice for inference, we limit the length of these *Consecutive Paths* to be at most 3, which means $|j - i| \leq 2$. Note that this definition can be naturally extended to the intra-sentence case where $j = i$. We thus consider the intra-sentence case as a type of the Consecutive Path. A pair of entities can correspond to multiple consecutive paths since they can be mentioned more than once.

Multi-Hop Paths Another typical case for inter-sentence relation instances is the multi-hop relation (Yao et al., 2019; Zeng et al., 2020a). In such cases, the head and tail entities are far from each other in the document but can be connected through *bridge entities*, just like the entity *The Espoo Cathedral* in Figure 1 bridges *the EC Parish* and *Finland* in sentence 1 and 6.

For these cases, we start from the head entity, go through all the bridge entities, arrive at the tail entity, and select all the corresponding sentences in this route as a path. Formally, for the head entity e_h and the tail entity e_t, the multi-hop relation indicates that there exist a list of bridge entities e_{b_1}, ..., e_{b_k} such that (e_h, e_{b_1}), (e_{b_1}, e_{b_2}), ..., (e_{b_k}, e_t) form $k+1$ intra-sentence relations respectively in sentence $S_{p_1}, \ldots, S_{p_{k+1}}$. Following this route, we choose these $k+1$ sentences as the *Multi-Hop Path*. Given the discovery in §2 that most instances only needs 3 sentences, we restrict k to be at most 2, i.e., with only 1 or 2 bridge entities. It is possible to have several multi-hop paths for a certain pair with different lists of bridge entities.

Default Paths If neither of the aforementioned rules applies, we consider a rough estimate for the evidence with the most relevant sentences. We collect all pairs of sentences where one contains the head entity and the other contains the tail entity as *Default Paths*. Formally, let $\{S_{h_1}, \ldots, S_{h_p}\}$ and $\{S_{t_1}, \ldots, S_{t_q}\}$ denote the sets of sentences that contain the head entity e_h and the tail entity e_t, respectively. For this entity pair, we will have $p \times q$ Default Paths $\{S_{h_1}, S_{t_1}\}, \ldots, \{S_{h_p}, S_{t_q}\}$. Note that this type of paths is extracted only when no paths are found with the previous two patterns.

4 Comparing with Annotated Evidence

To demonstrate the effectiveness of our heuristic rules, we check the size of our path set on DocRED

	Path Recall	#Sent	#Path
C	71.7%	2.31	1.71
M	31.5%	3.14	2.35
C+M	80.5%	2.73	2.37
C+M+D	87.5%	2.69	2.27
document	-	8.00	-

Table 2: C, M and D stand for Consecutive Paths, Multi-hop Paths, and Default Paths, respectively. #Path and #Sent are the average path numbers and average sentence numbers in the union of all paths.

and their consistency with the gold supporting evidence. As mentioned in §2, the gold annotation acts as a collection of all related evidence, while each of our extracted paths represents one possible and minimal sentence set. Ideally, if the path set is sufficient, all connecting sentences between the entity pair should be successfully captured. In other words, they would be presented via various paths in our path set. Therefore, the union of paths is expected to be a superset of the supporting evidence. We use the **Coverage** of the supporting evidence to measure the *sufficiency* of our path set, which stands for the percentage of instances whose supporting evidence is fully covered by the union of our paths. Meanwhile, the total number of paths ($\#Path$) and union size of the paths ($\#Sent$) should also remain at a low standard, so as to avoid *redundancy*.

Table 2 shows the statistics of the path sets extracted via our rules. The Consecutive Paths form a strong baseline that covers 71.7% of instances. Combining the three types, up to 87.5% of instances from the supporting evidence are fully covered by our path sets. The main reason that C+M+D can not cover all the instances is that the supporting evidence annotated in DocRED includes all associated sentences, while C+M+D only find a sufficient set to identify the relation.

Meanwhile, notice that the union of the three types contains only 2.69 different sentences on average, which means that our methods can filter out up to 2/3 of the original text. Also, our method is computationally efficient since only 2.27 paths need to be modeled on average. This demonstrates that our methods form a sufficient and non-redundant estimate for the gold supporting evidence, drastically alleviating the negative impact of irrelevant information.

Model	Dev			Test
	Intra-F1	Inter-F1	F1	F1
CNN	51.87	37.58	43.45	42.26
BiLSTM	57.05	43.49	50.94	51.06
HIN-Glove	60.83	48.35	52.95	53.30
GAT	58.14	43.94	51.44	49.51
GCNN	57.78	44.11	51.52	51.62
EoG	58.90	44.60	52.15	51.82
AGGCN	58.76	45.45	52.47	51.45
LSR-Glove	60.83	48.35	55.17	54.18
GAIN-Glove	61.67	48.77	55.29	55.08
Paths+BiLSTM	**62.73**	**49.11**	**56.54**	**56.23**

Table 3: Model performance on DocRED.

5 Experiments

To further validate the sufficiency of our selected paths, we perform evaluation on DocRED by feeding the paths to an RE model. While previous works take entire documents as input, we replace the document with our selected paths regarding a given entity pair. Intuitively, if the paths can cover all crucial information in the document, we would expect comparable or better performance with identical model architecture, as our paths contain little irrelevant information and may help focus on a few key sentences.

Setup Given a pair of entities, all paths are first extracted as described in §3. Since each path corresponds to one possible connection of the head and tail entities, we predict the relations with each path independently and aggregate the results afterwards.

For every single path c, we concatenate all sentences in it as one segment $[\mathbf{w_1^c}, ..., \mathbf{w_m^c}]$, where the order of sentences is the same as in the original document. The segment is fed to a BiLSTM to obtain the contextual embeddings $[\mathbf{h_1^c}, ..., \mathbf{h_m^c}]$. The representation of an entity mention, which spans from the s-th word to the t-th word, is defined as $\mathbf{m_k^c} = \frac{1}{t-s+1}\sum_{j=s}^{t}\mathbf{h_j^c}$. The representation of an entity e_i^c with K mentions is computed as the average of the representations of its mentions: $\mathbf{e_i^c} = \frac{1}{K}\sum_k \mathbf{m_k^c}$. Then, we use a two-layer perceptron to calculate the probability of each relation r based on the current path c: $P_{ij}^c(r) = \sigma(F([e_i^c; e_j^c; |e_i^c - e_j^c|; e_i^c * e_j^c]))$, where $\sigma(\cdot)$ is the Sigmoid function and $F(\cdot)$ stands for the two-layer perceptron.

After obtaining the prediction of every path between a given entity pair, we aggregate the predicted results by selecting the most likely predictions: $P_{ij}(r) = \max_c P_{ij}^c(r)$.

We use the Glove-100 (Pennington et al., 2014)

embedding for the BiLSTM encoder with hidden size 256. Following previous works (Nan et al., 2020b), we report the F1 for intra- and inter-sentence entity pairs along with the overall F1 score as evaluation metrics.

Results We compare our methods with previous sequence-based models and graph-based models. All these models take the entire document as input. As shown in Table 3, our selected path with BiLSTM achieves 56.23% F1 on the test set, which outperforms the sequence-based models. Compared with the baseline BiLSTM, our model brings 5.68% and 5.62% improvement on intra- and inter-sentence entity pairs on the dev set, respectively.

Surprisingly, our simple method achieves a higher performance compared with graph-based models, which are more complex and also possess the ability to filter out irrelevant information. Combined with our path-selection scheme, a BiLSTM can perform 1.25% and 1.15% better on the dev and test set, respectively, compared to the SOTA graph-based model in the same situation. This may indicate that, while graph-based models have shown excellent abilities to focus on important information in a self-adaptive manner, it is more helpful to explicitly select from the document than to fully rely on graph-based models. With a simple filtering scheme inspired by human annotations, we can better explore the potentials of existing models and produce better results.

6 Discussion

So far we have shown from experiments the limited number of sentences required to deduce a relation instance. While the interesting results seem unconventional for Document RE, which features complex inter-sentence relations, it is worth mentioning that possible explanations exist in current works in related fields. The interdisciplinary outlooks may provide helpful insights for community members to understand the causes of the *three-sentences* phenomenon and revisit the problem of Document-level Relation Extraction.

Linguistic Perspective One likely cause of the discussed phenomenon is that the seemingly distant relations are not so difficult given their linguistic form. Stevenson (2006) mentions that a majority of inter-sentence relation instances are in fact due to *co-references* (anaphoric expressions or alternative descriptions). In these cases, relations could be considered to be described entirely within one sentence but with head or tail entities being referred to indirectly. Considering anaphoric expressions are likely to appear in surrounding sentences for the candidate mentions (Chowdhury and Zweigenbaum, 2013), these findings are directly in line with our observation that consecutive paths could support more than 70% relation instances, and provide evidence for *three-sentences* phenomenon.

Cognitive Perspective Another possible explanation is that the RE task is naturally defined within a limited amount of entities and context, given the nature of the human brain. It is widely believed that *Working Memory* (WM) (Baddeley, 1992) plays a vital role to store and manipulate information in inference tasks (Barreyro et al., 2012), but the capacity of separate information chunks in WM are often limited to 4 (Cowan, 2001). As we need to memorize all the separate entities in the inference chain along with their relations, it is natural that we tend to describe a relation within a limited number of sentences, since rendering a relationship with more sentences may cause our WM to exceed its capacity. Daneman and Carpenter (1980) show that the success rate of completing a reading task drastically drops if too much information, exceeding the subject's WM capacity, is required for the task. Therefore, as the datasets are constructed from natural language, the *three-sentences* phenomenon in the data may be a common pattern that we (unconsciously) follow for mutual understanding.

7 Conclusion

In this paper, we perform an analysis over 3 document RE benchmark datasets, and find that human annotators often use a small number of sentences to extract entity relations in document level. This motivates us to think over which sentences are critical for document RE. We carefully design heuristic rules to select informative *path* sets from entire documents, which can be further combined with a simple BiLSTM to achieve competitive performance on a benchmark dataset, even better than complex graph-based methods.

Acknowledgments

We thank the anonymous reviewers for the helpful comments and suggestions. This work is supported in part by the National Hi-Tech R&D Program of China (2018YFC0831900) and the NSFC Grants (No.61672057, 61672058).

References

Alan Baddeley. 1992. Working memory. *Science*, 255(5044):556–559.

Juan Pablo Barreyro, Jazmín Cevasco, Débora Burín, and Carlos Molinari Marotto. 2012. Working memory capacity and individual differences in the making of reinstatement and elaborative inferences. *The Spanish journal of psychology*, 15(2):471.

Md Faisal Mahbub Chowdhury and Pierre Zweigenbaum. 2013. A controlled greedy supervised approach for co-reference resolution on clinical text. *Journal of biomedical informatics*, 46(3):506–515.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China. Association for Computational Linguistics.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114.

Meredyth Daneman and Patricia A Carpenter. 1980. Individual differences in working memory and reading. *Journal of verbal learning and verbal behavior*, 19(4):450–466.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative V CDR task corpus: A resource for chemical disease relation extraction. *Database: the journal of biological databases and curation*, 2016:baw068.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020a. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020b. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182, Valencia, Spain. Association for Computational Linguistics.

Mark Stevenson. 2006. Fact distribution in information extraction. *Language resources and evaluation*, 40(2):183–201.

Kumutha Swampillai and Mark Stevenson. 2010. Inter-sentential relations in information extraction corpora. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Ye Wu, Ruibang Luo, Henry CM Leung, Hing-Fung Ting, and Tak-Wah Lam. 2019. Renet: A deep learning approach for extracting gene-disease associations from literature. In *International Conference on Research in Computational Molecular Biology*, pages 272–284. Springer.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020a. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020b. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Unsupervised Cross-Domain Prerequisite Chain Learning using Variational Graph Autoencoders

Irene Li, Vanessa Yan, Tianxiao Li, Rihao Qu and **Dragomir Radev**
Yale University, USA
{irene.li,vanessa.yan,tianxiao.li,rihao.qu,dragomir.radev}@yale.edu

Abstract

Learning prerequisite chains is an essential task for efficiently acquiring knowledge in both known and unknown domains. For example, one may be an expert in the natural language processing (NLP) domain but want to determine the best order to learn new concepts in an unfamiliar Computer Vision domain (CV). Both domains share some common concepts, such as machine learning basics and deep learning models. In this paper, we propose unsupervised cross-domain concept prerequisite chain learning using an optimized variational graph autoencoder. Our model learns to transfer concept prerequisite relations from an information-rich domain (source domain) to an information-poor domain (target domain), substantially surpassing other baseline models. Also, we expand an existing dataset by introducing two new domains—CV and Bioinformatics (BIO). The annotated data and resources, as well as the code, will be made publicly available.

1 Introduction

With the rapid growth of online educational resources in diverse fields, people need an efficient way to acquire new knowledge. Building a concept graph can help people design a correct and efficient study path (ALSaad et al., 2018; Yu et al., 2020). There are mainly two approaches to learning prerequisite relations between concepts: one is to extract the relations directly from course content, video sequences, textbooks, or Wikipedia articles (Yang et al., 2015b; Pan et al., 2017; Alzetta et al., 2019), but this approach requires extra work on feature engineering and keyword extraction. Our method follows a different approach of inferring the relations within a concept graph (Liang et al., 2018; Li et al., 2019, 2020).

In a concept graph, we define $p \rightarrow q$ as the notion that learning concept p is a prerequisite to learning concept q. Existing methods formulate this question as a classification task. A typical method is to encode concept pairs and train a classifier to predict if there is a prerequisite relation (Alzetta et al., 2019; Yu et al., 2020). However, this method requires annotated prerequisite pairs during training. Alternatively, others have used graph-based models to predict prerequisite relations. Gordon et al. (2016) proposed information-theoretic approaches to infer concept dependencies. Li et al. (2019) modeled a concept graph using Variational Graph Autoencoders (VGAE) (Kipf and Welling, 2016), training their model to infer unseen prerequisite relations in a semi-supervised way. While most of the previous methods were supervised or semi-supervised, Li et al. (2020) introduced Relational-VGAE, which enabled unsupervised learning on prerequisite relations.

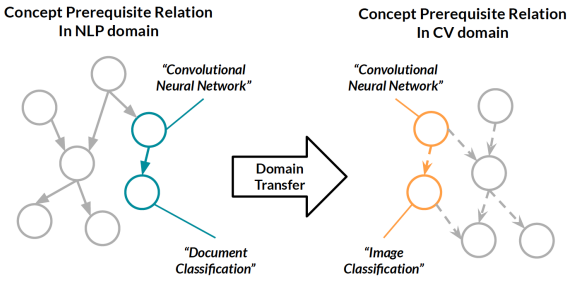


Figure 1: Cross-domain prerequisite chains.

Existing work mainly focuses on prerequisite relations within a single domain. In this paper, we tackle the task of cross-domain prerequisite chain learning, by transferring prerequisite relations between concepts from a relatively information-rich domain (source domain) to an information-poor domain (target domain). As an example, we illustrate in Figure 1, a partial concept graph from the Natural Language Processing (NLP) domain and a partial concept graph from the Computer Vision (CV) domain. Prerequisite relations among concepts in the NLP domain are known, and we seek to infer prerequisite relations among concepts in

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1005–1011
August 1–6, 2021. ©2021 Association for Computational Linguistics

the CV domain. These two domains share some concepts, such as *Convolutional Neural Network*. We assume that being aware of prerequisite relations among concepts in the source domain helps infer potential relations in the target domain. More specifically, in the figure, knowing that *Convolutional Neural Network→Document Classification* helps us determine that *Convolutional Neural Network→Image Classification*.

Our contributions are two-fold. First, we propose cross-domain variational graph autoencoders to perform unsupervised prerequisite chain learning in a heterogeneous graph. Our model is the first to do domain transfer within a single graph, to the best of our knowledge. Second, we extend an existing dataset by collecting and annotating resources and concepts in two new target domains. Data and code will be made public in https://github.com/Yale-LILY/LectureBank/tree/master/LectureBankCD.

2 Dataset

LectureBank2.0 (Li et al., 2020) dataset contains 1,717 lecture slides (hereon called **resources**) and 322 concepts with annotated prerequisite relations, largely from NLP. We treat this dataset as our information-rich source domain (NLP). Also, we propose an expansion dataset, LectureBankCD, by introducing two new target domains in the same data format: CV and Bioinformatics (BIO). We report statistics on the dataset in Table 1. For each domain, we identify high-quality lecture slides from the top university courses, collected by domain experts, and we choose concepts by crowd-sourcing. We end up with 201 CV concepts and 100 BIO concepts. In each domain, we ask two graduate-level annotators with deep domain knowledge to add prerequisite chain annotations for every possible pair of concepts. The Cohen's kappa agreement scores (McHugh, 2012) are 0.6396 for CV and 0.8038 for BIO. Cohen's kappa between 0.61–0.80 is considered substantial, so our annotations are reliable.

Domain	Files	Pages	Tks/pg	Con.	PosRel
NLP	1,717	65,028	47	322	1,551
CV	1,041	58,32	43	201	871
BIO	148	7,13	135	100	234

Table 1: LectureBankCD statistics on NLP, CV and BIO domain: Tks/pg (Tokens per slide page), Con. (Number of concepts), PosRel (Positive Relations).

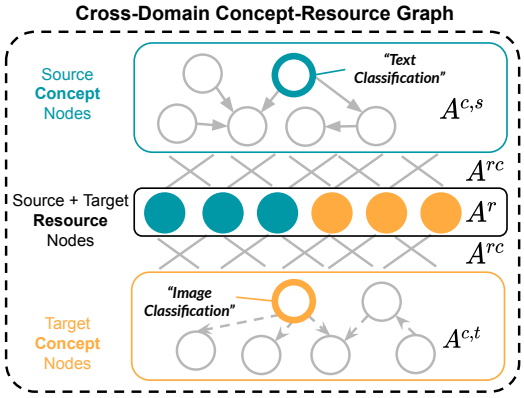


Figure 2: Cross-Domain Concept-Resource Graph: we model the resource nodes (solid nodes) and concept nodes (hollow nodes) from two domains (in blue and orange) in a heterogeneous graph. We show a subset of nodes and edges.

We take the union of the positive annotations for our experiments: 871 positive relations for CV and 234 positive relations for BIO.

3 Methodology

Inspired by Li et al. (2020), we build a cross-domain concept-resource graph $G = (X, A)$ that includes resource nodes and concept nodes from both the source and target domains (Figure 2). To obtain the node feature matrix X, we use either BERT (Devlin et al., 2019) or Phrase2Vec (Artetxe et al., 2018) embeddings. We consider four edge types to build the adjacency matrix A: $A^{c,s}$: edges between source concept nodes; A^{rc}: edges between all resource nodes and concept nodes; A^{r}: edges between resource nodes only; and $A^{c,t}$: edges between target concept nodes. In unsupervised prerequisite chain learning, $A^{c,s}$—concept relations of the source domain—are known, and the task is to predict $A^{c,t}$—concept relations of the target domain. For A^{rc} and A^{r}, we calculate cosine similarities based on node embeddings, consistent with previous works (Li et al., 2019; Chiu et al., 2020).

Cross-Domain Graph Encoder VGAE (Kipf and Welling, 2016) contains a graph neural network (GCN) encoder (Kipf and Welling, 2017) and an inner product decoder. In a GCN, the hidden representation of a node i in the next layer is computed using only the information of direct neighbours and the node itself. To account for cross-domain knowledge, we additionally consider the *domain neighbours* for each node i. These domain neighbours are a set of common or semantically similar

concepts from the other domain.[1] We define the cross-domain graph encoder as:

$$h_i^{(l+1)} = \sigma\Bigg(\sum_{j \in N_i} W^{(l)} h_j^{(l)} + W^{(l)} h_i^{(l)} + \sum_{k \in N_i^D} W_D^{(l)} h_k^{(l)}\Bigg)$$

where N_i denotes the set of direct neighbours of node i, N_i^D is the set of domain neighbours, and W_D and W are trainable weight matrices. To determine the domain neighbors, we compute cosine similarities and match the concept nodes only from source domain to target domain: $cosine(h_s, h_t)$. The values are then normalized into the range of [0,1], and we keep the top 10% of domain neighbors.[2]

DistMult Decoder We optimize the original inner product decoder from VGAE. To predict the link between a concept pair (c_i, c_j), we apply the DistMult (Yang et al., 2015a) method: we take the output node features from the last layer, $\hat{X}$, and define the following score function to recover the adjacency matrix $\hat{A}$ by learning a trainable weight matrix R: $\hat{A} = \hat{X}R\hat{X}$. A Sigmoid function is used to predict positive/negative labels from $\hat{A}$.

4 Evaluation

We evaluate on our new corpus LectureBankCD, treating the NLP domain as the source domain and transferring to the two new target domains: NLP→CV and NLP→BIO. Consistent with Kipf and Welling (2017); Li et al. (2019), we randomly split the positive relations into 85% training, 5% validation, and 10% testing. To account for imbalanced data, we randomly select negative relations such that the training set has the same number of positive and negative relations. We do the same for the validation and test sets. We report average scores over five different randomly seeded splits.

To encode concepts and resources, we test BERT and P2V embeddings. For BERT, we applied a pre-trained version from Google[3]. We trained P2V using all the resource data. Both methods only require free-text for training and encoding.

Baseline Models We concatenate the BERT/P2V embeddings of each pair of concepts and feed the result into a classifier (CLS + BERT and CLS + P2V). We train the classifier on the source domain only, then evaluate on the target domain. We report the best performance among Support Vector Machine, Logistic Regression, Gaussian Naïve Bayes, and Random Forest. In addition, we train the VGAE model Li et al. (2019) on the source domain and test on the target domain, initializing the VGAE input with BERT and P2V embeddings separately (VGAE + BERT and VGAE + P2V). Given that GAE is structurally similar to VGAE, we leave this for future work. Other graph-based methods including DeepWalk (Perozzi et al., 2014) and Node2vec (Grover and Leskovec, 2016) are not applicable in this setting as both models require training edges from the target domain in order to generate node embeddings for target concepts.

Proposed Method We report results of our proposed model, CD-VGAE, initialized with BERT and P2V node embeddings separately. Consistent with the work from Li et al. (2019) and Li et al. (2020), P2V embeddings yield better results than BERT embeddings in general. One possible reason for this difference is that BERT embeddings have a large number of dimensions, making it very easy to overfit. The two CLS models yield a negative result, with F1 worse than random guess. A possible reason is that treating concept pairs independently from the source domain may not be beneficial for the target domains. The VGAE models have a better performance when considering the concepts in a large graph. As shown in the table, our method performs better than the chosen baselines on both accuracy and F1 score, by incorporating information from domain neighbors. In particular, it yields much higher recall than all the baseline models. We provide further analysis in a later section.

Upper Bound Performance Finally, we conduct in-domain experiments on CV and BIO (supervised training and testing in the target domain), to show an upper bound for cross-domain performance. We test a variety of methods including traditional classifiers as well as graph-based approaches, including DeepWalk, Node2vec, and GraphSAGE (Hamilton et al., 2017).

5 Analysis

Next, we conduct quantitative analysis and case studies on the target domain concept graphs recovered by our model (CD-VGAE+P2V) and two

[1] In Figure 2, the two labeled nodes are domain neighbors.
[2] Parameter is selected using validation dataset.
[3] https://github.com/google-research/bert, (version with L = 12 and H = 768)

Method	NLP→CV F1	Acc	Pre	Rec	NLP→BIO F1	Acc	Pre	Rec
Baseline Models								
CLS + BERT	0.4277	0.5480	0.5743	0.3419	0.3930	0.6000	0.7481	0.2727
CLS + P2V	0.4881	0.5757	0.6106	0.4070	0.2222	0.5333	0.6000	0.1364
VGAE + BERT (Li et al., 2019)	0.5885	0.5477	0.5398	0.6488	0.6011	0.6091	0.6185	0.5909
VGAE + P2V (Li et al., 2019)	0.6202	0.5500	0.5368	0.7349	0.6177	0.6273	0.6521	0.6091
Proposed Method								
CD-VGAE + BERT	0.6391	0.5593	0.5441	0.7884	0.6289	0.6273	0.6425	0.6364
CD-VGAE + P2V	**0.6754**	**0.5759**	0.5468	0.8837	**0.6512**	**0.6591**	0.6667	0.6364
Supervised Performance – Upper Bound								
CLS + Node2vec (Grover and Leskovec, 2016)	0.8172	0.8197	0.8223	0.8140	0.8060	0.7956	0.7547	0.8727

Table 2: Evaluation results on two target domains. Underlined scores are the best among the baseline models.

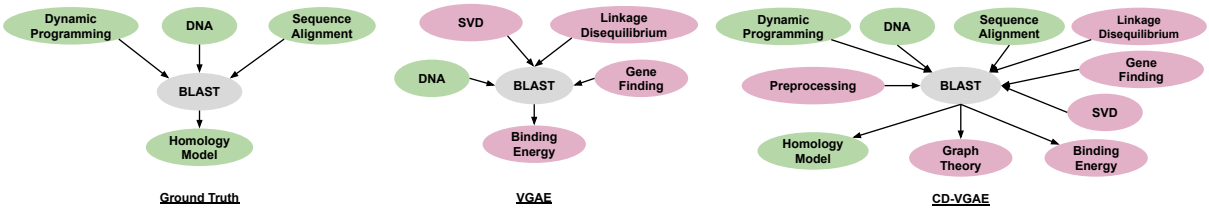


Figure 3: Case Study in BIO: direct neighbors of *BLAST*, including successors and prerequisites, from the ground truth, VGAE, and our proposed CD-VGAE model. SVD stands for Singular Value Decomposition. Correct nodes are marked in blue, incorrect nodes are marked in red. (Best viewed in color!)

baseline models (CLS + P2V, VGAE + P2V), to take a closer look at the results.

Quantitative Analysis We first apply the three trained models to recover the concept graph in the CV domain. Compared to the ground truth with 871 positive relations, the baseline model predicts 527, VGAE predicts 963, and our model predicts 1,209. Similarly, in the BIO domain with 234 positive relations, the baseline model predicts only 128 positive edges, VGAE predicts 261, and our model predicts 303. Since our model tends to predict more positive edges, it has a higher recall. A higher recall is preferred in real-world applications as a system should not miss any relevant concepts when designing a user's study path.

Concept Graph Recovery We now provide case studies of the recovered concept graphs. In Table 3, we show successors of the concept *Image Processing* from the CV domain, i.e. concepts for which *Image Processing* is a prerequisite. Both the baseline model and VGAE miss many successor concepts, whereas our model can recover a correct list without any missing concepts.

We illustrate another case study from the BIO domain in Figure 3 using the concept *BLAST* (short for "basic local alignment search tool "), an algorithm for comparing primary biological sequence information. In the ground truth, BLAST has three prerequisite concepts (*Dynamic Programming*, *DNA* and *Sequence Alignment*), and one successor concept (*Homology Model*). We observe that VGAE predicts only one prerequisite, *DNA*, and misses all the others. In contrast, our model successfully includes all the ground truth relations, although it predicts some extra ones compared to VGAE. A closer look at the extra predictions reveals that these are still relevant topics, even though they are not direct prerequisites. For example, *Sequence Alignment*, *BLAST* and *Graph Theory* are all associated with sequence analysis and share some common algorithms (i.e. De Bruijn Graph).

We provide a case study in the CV domain, shown in Figure 4, by selecting concept node *Object Localization*. The ground truth shows that it has 14 direct neighbors. The VGAE model only predicts five neighbors, while our model predicts more. Our model has two wrong predictions, but it gets 12 correct ones. In contrast, the VGAE model misses up to 10 neighbors, which is not acceptable in an application scenario of an educational platform leading students to miss very useful information.

6 Conclusion

In this paper, we proposed the CD-VGAE model to solve the task of cross-domain prerequisite chain

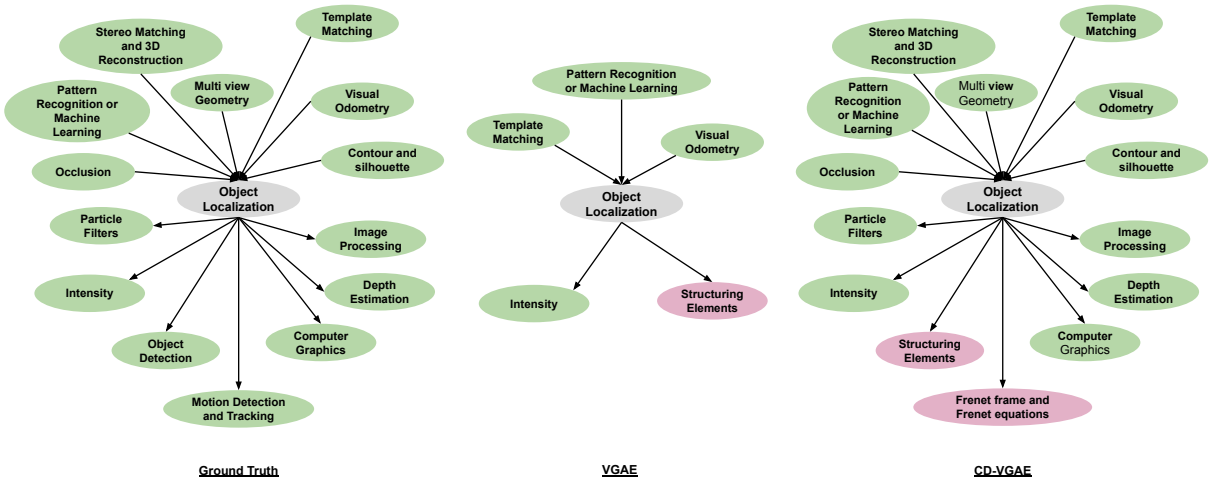


Figure 4: Case Study in CV: direct neighbors of *Object Localization*.

Base	**VGAE**
Image Representation OCR	Image Representation Computer graphics Eye Tracking
CD-VGAE	**Ground Truth**
Video/Image augmentation Image Representation Face Detection Emotion Recognition Feature Extraction Feature Learning OCR Computer Graphics Eye Tracking	Video/Image augmentation Image Representation Face detection Emotion Recognition Feature Extraction Feature Learning OCR Computer Graphics Eye Tracking

Table 3: Successors of the concept *Image Processing*, i.e. concepts for which *Image Processing* is a prerequisite (OCR stands for Optical Character Recognition).

learning. Results show that our model outperforms previous unsupervised graph-based models by a large margin, especially with respect to the F1 and recall scores. In addition, we created a new dataset that contains resources and concepts from two domains along with annotated prerequisite relations.

References

Fareedah ALSaad, Assma Boughoula, Chase Geigle, Hari Sundaram, and ChengXiang Zhai. 2018. Mining mooc lecture transcripts to construct concept dependency graphs. *International Educational Data Mining Society*.

Chiara Alzetta, Alessio Miaschi, Giovanni Adorni, Felice Dell'Orletta, Frosina Koceva, Samuele Passalacqua, and Ilaria Torre. 2019. Prerequisite or not prerequisite? that's the problem! an nlp-based approach for concept prerequisite learning. In *CLiC-it*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Billy Chiu, Sunil Kumar Sahu, Derek Thomas, Neha Sengupta, and Mohammady Mahdy. 2020. Autoencoding keyword correlation graph for document clustering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3974–3981, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–875, Berlin, Germany. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Irene Li, Alexander Fabbri, Swapnil Hingmire, and Dragomir Radev. 2020. R-VGAE: Relational-variational graph autoencoder for unsupervised prerequisite chain learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1147–1157, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Irene Li, Alexander R. Fabbri, Robert R. Tung, and Dragomir R. Radev. 2019. What should I learn first: Introducing lecturebank for NLP education and prerequisite chain learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6674–6681. AAAI Press.

Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C. Lee Giles. 2018. Investigating active learning for concept prerequisite learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 7913–7919. AAAI Press.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Prerequisite relation learning for concepts in MOOCs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, Vancouver, Canada. Association for Computational Linguistics.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710. ACM.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015a. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yiming Yang, Hanxiao Liu, Jaime G. Carbonell, and Wanli Ma. 2015b. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 159–168. ACM.

Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. 2020. MOOCCube: A large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, Online. Association for Computational Linguistics.

A Supervised Results

Method	Acc	F1	Pre	Rec
GS+BERT	0.7491	0.7513	0.7404	0.7628
GS+P2V	0.7457	0.7423	0.7486	0.7372
CLS+P2V	0.7642	0.757	0.7754	0.7395
CLS+BERT	0.7572	0.7495	0.7677	0.7326
DeepWalk	0.7988	0.791	0.8182	0.7674
Node2vec	**0.8197**	**0.8172**	0.8223	0.8140

Table 4: Supervised evaluation results: CV→CV. GS:GraphSAGE.

Method	Acc	F1	Pre	Rec
GS+BERT	0.7289	0.7355	0.7104	0.7727
GS+P2V	0.7911	0.7904	0.7787	0.8091
CLS+P2V	0.72	0.7367	0.6874	0.8091
CLS+BERT	0.7067	0.7189	0.683	0.7727
DeepWalk	0.7911	0.8079	0.7334	0.9091
Node2vec	**0.7956**	**0.8060**	0.7547	0.8727

Table 5: Supervised evaluation results: BIO→BIO. GS:GraphSAGE.

As a supplementary experiment, we present in-domain results in Table 4, 5: CV→CV and BIO→BIO respectively. While we show in the main paper that CLS + Node2vec yields the best result, which serves as an upper bound on cross-domain performance, we additionally show our experimental results for other supervised methods:

CLS + P2V/BERT We encode concept pairs with P2V/BERT, concatenate the embeddings of both concepts within each possible pair, and then train a binary classifier. We report the best performance among Support Vector Machine, Logistic Regression, Gaussian Naïve Bayes, and Random Forest.

DeepWalk, Node2vec DeepWalk (Perozzi et al., 2014) randomly samples a node and traverses to a neighbor node until it reaches a maximum length, updating the latent representation of each node after each "walk "to maximize the probability of each node's neighbors given a node's representation. Node2Vec (Grover and Leskovec, 2016) improves DeepWalk by providing the additional flexibility of placing weights on random walks. For both methods, we input the training prerequisite relations and obtain concept node embeddings. After generating embeddings for each concept in the target domain, we concatenate the embeddings of both concepts in each concept pair and pass the concatenated representation into a classifier to predict the relation. Again, we report the best performance from the same four classifiers.

GraphSAGE + P2V/BERT GraphSAGE (Hamilton et al., 2017) is an inductive framework to generate node embeddings for unseen data by leveraging existing node features. We first treat it as a node embedding method, as done with DeepWalk and Node2vec. After generating concept node embeddings, we train a classifier to predict concept relations and report in-domain results. In addition, we investigate GraphSAGE for the out-of-domain setting. We assume that, because there are unseen topics when transferring to new domains, such an inductive method like GraphSAGE may fit in our scenario. However, we end up with negative results as the original GraphSAGE may not fit in to this specific application. We leave further investigation for future work.

Attentive Multiview Text Representation for Differential Diagnosis

Hadi Amiri[a,c], **Mitra Mohatarami**[b], **Isaac S. Kohane**[c]
[a]Department of Computer Science, University of Massachusetts, Lowell
[b]MIT Computer Science and Artificial Intelligence Laboratory
[c]Department of Biomedical Informatics, Harvard University
Massachusetts, USA
hadi_amiri@uml.edu, mitram@mit.edu, isaac_kohane@harvard.edu

Abstract

We present a text representation approach that can combine different views (representations) of the same input through effective data fusion and attention strategies for ranking purposes. We apply our model to the problem of *differential diagnosis*, which aims to find the most probable diseases that match with clinical descriptions of patients, using data from the Undiagnosed Diseases Network. Our model outperforms several ranking approaches (including a commercially-supported system) by effectively prioritizing and combining representations obtained from traditional and recent text representation techniques. We elaborate on several aspects of our model and shed light on its improved performance.

1 Introduction

Electronic Health Records (EHRs) (Dick et al., 1997) contain a wealth of documented information and insights about patients health and well-being. However, it is difficult to effectively process such data due to complex terminology, missing information, and imprecise clinical descriptions (Friedman et al., 2013; Rajkomar et al., 2019). In addition, an especially challenging class of diseases are orphan or rare diseases (Kodra et al., 2012; Walley et al., 2018), which are diverse in symptoms and affect a smaller percentage of the population.

In this paper, we investigate how well Natural Language Processing (NLP) algorithms could reproduce the performance of clinical experts in the task of *differential diagnosis*–the process of distinguishing a particular disease from others that present similar clinical features, given medical histories (descriptions) of individual patients. We formulate this task as a ranking problem where the aim is to find the most probable diseases given medical histories of patients (Dragusin et al., 2013).

We develop a novel *pairwise* ranking algorithm that combines different views of patient and disease descriptions, and prioritizes effective views through an **A**ttentive **M**ultiview **N**eural **M**odel (AMNM). We research this problem using data from the Undiagnosed Diseases Network (UDN) (Gahl et al., 2015; Ramoni et al., 2017)[1], which includes concise medical history of patients and their corresponding diseases in the Online Mendelian Inheritance in Man (OMIM) dataset (Amberger et al., 2015).[2] All diagnoses–mappings between each patient and corresponding diseases–are provided by a team of expert clinicians from the UDN.

The contributions of this paper are as follows:

- illustrating the impact of NLP in detecting the nature of illness (diagnosis) in patients with rare diseases in a real-world setting, and
- a novel neural approach that effectively combines and prioritizes different views (representations) of inputs for ranking purposes.

Our Attentive Multiview Neural Model employs traditional and recent representation learning techniques and outperforms current pairwise neural ranking approaches through effective data fusion and attention strategies. We conduct several experiments to illustrate the utility of different fusion techniques for combining patient (query) and disease (document) representations.[3]

2 Method

In many domains, entities can be represented from multiple views. For example, a patient can be represented by demographic data, medical history, diagnosis codes, radiology images, etc. We propose a neural model to effectively prioritize important views and combine them for ranking purposes.

[1] https://undiagnosed.hms.harvard.edu/
[2] https://www.omim.org/
[3] code: https://clu.cs.uml.edu/tools.html

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1012–1019
August 1–6, 2021. ©2021 Association for Computational Linguistics

Figure 1 shows our model, which comprises of three major components: (a): an attention network that estimates and weights the contribution of each view in the ranking process, (b): a fusion network the utilizes intra-view feature interactions to effectively combine query-document representations, and (c): a `softmax` layer at the end that estimates the query-document relevance scores given their combined representations. We first formulate the problem and then explain these components.

2.1 Problem Statement

Let $(\mathbf{q}', \mathbf{d}')$ and $(\mathbf{q}'', \mathbf{d}'')$ denote two different views of the same query and document (throughout the paper, we think of queries and documents as clinical descriptions of patients and diseases respectively).[4] These views can be obtained using traditional (Robertson and Walker, 1994) or recent (Devlin et al., 2019) representation learning techniques applied to textual descriptions or codified data of queries and documents. For example, $\mathbf{q}'$ and $\mathbf{d}'$ can indicate representations of the *texts* of a query and a document, and $\mathbf{q}''$ and $\mathbf{d}''$ can indicate representations of the *medical concepts and codes* associated with the same query and document. Our task is to determine a relevance score between each given query and document. Toward this goal, we effectively prioritize and combine these representations through Attention and Fusion neural networks, which are described blow.

2.2 Attention Model

We develop an attention sub-network to explicitly capture the varying importance of views by assigning attentive weights to them. Specifically, given the embedding vectors of a query $\mathbf{q}^i \in \mathbb{R}^l$ and a document $\mathbf{d}^i \in \mathbb{R}^m$ in the ith view, we use a Feed-forward network, i.e. function $f(.)$ in Figure 1, to estimate the vector $\mathbf{a}$ that captures attention weights across views as follows:

$$f(\mathbf{q}^i, \mathbf{d}^i) = \varphi(\mathbf{W}^q \mathbf{q}^i + \mathbf{b}^q)^\top \cdot \varphi(\mathbf{W}^d \mathbf{d}^i + \mathbf{b}^d),$$
$$\mathbf{a} = \texttt{softmax}([f(\mathbf{q}^i, \mathbf{d}^i), \forall i]), \tag{1}$$

where $\mathbf{W}^q \in \mathbb{R}^{n \times l}$ and $\mathbf{W}^d \in \mathbb{R}^{n \times m}$ are weight matrices to transform the query and document representations into the same underlying space of dimension n, $\mathbf{b}^q \in \mathbb{R}^n$ and $\mathbf{b}^d \in \mathbb{R}^n$ are the trainable bias vectors for the query and document respectively and $\varphi(.)$ is the ReLU function. The

[4] Our model can incorporate any number of views; we only illustrate two views here for simplicity.

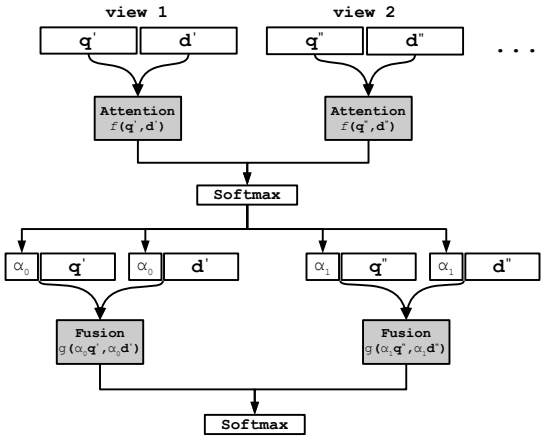


Figure 1: The architecture of our Attentive Multiview Neural Model (AMNM). For simplicity, we illustrate two views only, e.g. $(\mathbf{q}', \mathbf{d}')$ indicates representations of the texts of a query and a document, and $(\mathbf{q}'', \mathbf{d}'')$ indicates representations of the medical codes and concepts associated with the same query and document. $f(.)$ and $g(.)$ indicate attention and fusion functions respectively, and $\mathbf{a}_i$ indicates the attentive weight of the ith view estimated by the attention sub-network.

`softmax` activation function transforms the attention weights to $[0, 1]$ range. Assuming that the query-document pair of the more influential view are more similar in the underlying shared space (estimated by dot product in (1)), $\mathbf{a}$ captures attention weights of different views.

2.3 Fusion Model

Previous learning to rank approaches often concatenate query and document representations to combine their corresponding features (dos Santos et al., 2015; Amiri et al., 2016). There are a few approaches that explicitly capture feature interactions between queries and documents (Severyn and Moschitti, 2015; Echihabi and Marcu, 2003). We extend these fusion techniques and compare them.

Given the attention weights from (1), we develop a fusion sub-network, function $g(.)$ in Figure 1, to capture the intra-view feature interactions for query and document representations of each view. Our fusion network takes as input the *attentive embeddings* of each view, i.e. $(\alpha \times \mathbf{q}, \alpha \times \mathbf{d})$, and combines them through *one* of the following tensor fusion operations:

$$\begin{aligned} g^{dot}(\alpha\mathbf{q}, \alpha\mathbf{d}) &= \alpha^2 \times \varphi(\mathbf{W}^q\mathbf{q} + \mathbf{b}^q)^\top \cdot \\ &\quad \varphi(\mathbf{W}^d\mathbf{d} + \mathbf{b}^d), \\ g^{outer}(\alpha\mathbf{q}, \alpha\mathbf{d}) &= \alpha^2 \times \mathbf{q} \otimes \mathbf{d}, \\ g^{conv}(\alpha\mathbf{q}, \alpha\mathbf{d}) &= \alpha^2 \times \texttt{Conv1d}(\mathbf{q} \otimes \mathbf{d}), \end{aligned} \tag{2}$$

where g^{dot}, g^{outer}, and g^{conv} denote the dot product, outer product, and one-dimensional (1D) convolution with average pooling. In contrast to g^{dot}, g^{outer} and g^{conv} are considerably more expensive operations but may better encode feature interactions. The output of function g is flattened and considered as the *intra-view embedding*.

Finally, we obtain the overall fused representation for each view by concatenating its intra-view and attentive embeddings. The representations of all views are then fed into a `softmax` to estimate the relevance between queries and documents.

3 Experiments

Data: Our data includes medical histories of 257 patients provided by the the Undiagnosed Diseases Network (UDN[5]) (Gahl et al., 2015; Ramoni et al., 2017), as well as general descriptions (including clinical features) of more than 9K diseases available in the Online Mendelian Inheritance in Man (OMIM) dataset (Amberger et al., 2015). The UDN is a nationwide program that improves the level of diagnosis for individual patients (with severe clinical conditions) whose signs and symptoms have been intractable to diagnosis (Kobren et al., 2021; Amiri et al., 2021). To the best of our knowledge, this dataset is the largest available dataset for investigation on rare disease patients. The relevance judgment between patients and diseases is provided by a team of expert clinicians at the UDN. The total number of positive patient-disease pairs is $4,746$, where the number of unique diseases among these pairs is $1,131$; note that different patients can match with the same disease. We split the patients into training (80%), validation (10%), and test (10%) sets. In addition, for each positive pair in the training set, we create a negative pair for the same patient through random sampling of diseases. At test time, we create all the possible patient-disease pair combinations (more than 218K pairs) and use the estimated confidence scores of the classifier to rank all diseases against each test patient. In terms of views, we consider the texts of medical histories and diseases as the first view, and medical concepts and codes extracted from histories by QuickUMLS (Soldaini and Goharian, 2016) as the second view.

We note the concept and code view provides a higher level and more general semantic distinctions by grouping semantically-similar terms, while text view encodes other elements of semantics such as negation, hedging, etc.

Baselines: We consider the following baselines:

• **BM25** (Robertson et al., 1995): An unsupervised approach that effectively predicts relevance based on term frequency, inverse document frequency, and document length.

• **SVMs** (Cortes and Vapnik, 1995): We develop TF/IDF weighted ngrams (n=[1–2]) as features for the text and code/concept views, and conduct exhaustive search over hyperparameters for best performance on validation data. Such features were found effective on clinical texts by previous work (Howes et al., 2012; Reuber et al., 2009).

• **BERT** (Devlin et al., 2019): An attentive bidirectional language model that estimates the relevance between queries and documents by generating contextual representations, jointly conditioned on left and right contexts. We use BERT models developed for clinical text (Alsentzer et al., 2019).[6]

• **SVM**^{rank} (Joachims, 2002): An extension of SVMs to ranking problems which adaptively sorts documents based on their relevance to each query through empirical risk minimization. As features, we use relevance scores or probability predictions generated by the above baselines as well as additional features (unigram overlap and IDF-weighted unigram overlap) (Yu et al., 2014) to better establish the relevance between queries and documents.

• **PhenoTips** (Girdea et al., 2013): This *commercial* tool is currently used at the UDN to assist diagnostic efforts. It utilizes external sources such as the Human Phenotype Ontology (Köhler et al., 2017) and Orphanet data[7] to rank candidate diseases according to their ontology-based similarity to phenotypic descriptions of patients. PhenoTips employs advanced statistical modeling to differentiate candidate diseases, accounts for disorder frequencies in the general population according to Orphanet, supports negative phenotypes–symptoms that were not observed in the patient–and utilizes both code and text views.

[5]Access to phenotypic and genomic UDN data can be granted by submitting an online access request at dbGaP: https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs001232.v1.p1.

[6]We input medical concepts to BERT by replacing them with their "preferred" concept, determined by UMLS (Lindberg, 1990; Bodenreider, 2004), across all patient and disease descriptions. For example, "diabetes mellitus type 1," "type 1 diabetes," "juvenile diabetes" and "IDDM" are all converted to "juvenile diabetes" (as the preferred concept).

[7]http://www.orpha.net

Settings: Initial representations for patient and disease descriptions are obtained from clinical BERT (Devlin et al., 2019; Alsentzer et al., 2019), i.e. $d_1, d_2 = 768$. In (1) and (2), we set the dimension of the shared space between query and document representations to $n = 100$. In addition, for the CNN fusion model, see (2), we use 250 filters and kernel size of 3. Further details are provided in the supplementary materials.

Evaluation Metrics: We employ Mean Average Precision (MAP), Precision at rank K (P@K), and Precision-Recall curve implemented in `trec_eval`[8] to compare competing systems. We use t-test for significance testing and asterisk mark (*) to indicate significant difference at $\rho = 0.01$.

3.1 Experimental Results

We report the performance of single and multiview models separately to ease comparison between views. The overall MAP and P@K, $\forall$K $\in \{5, 10\}$, performance of baselines for each view are reported in Table 1. The results show that BERT outperforms the other baselines across almost all measures. We attribute the poor performance of BM25 and SVMs to considerable difference in the underlying word/concept distribution in query and document spaces which can't be effectively addressed through lexical features (Burgun and Bodenreider, 2001; Pedersen et al., 2007).[9] In addition, BERT (code view) shows lower performance than BERT (text view). We conjecture that this results could be explained through the following points: (a): BERT is a strong language model and is robust in retrieving noun hypernyms or in completions involving shared category or role reversal (Ettinger, 2020), and (b): replacing medical concepts in text with their preferred concepts (see footnote 6) makes the original text less coherent, which can adversely affect the performance of BERT.

Table 2 shows the performance of SVM^{rank} with combined features across views, PhenoTips, and our Attentive Multiview Neural Model (AMNM) with different fusion functions. AMNM combines traditional and recent representation learning techniques by using BERT representations for text view, and BERT and SVMs representations for code view. All model combinations except for AMNM$_{bert\text{-}svms}$ (g^{conv}) lead to significant improvement against the best performing baseline–BERT (text view) in Table 1. AMNM$_{bert\text{-}bert}$ (g^{dot}) improves the best baseline by 3.4, 1.7 and 5.8 points in MAP, P@5 and P@10 respectively; the corresponding improvement for AMNM$_{bert\text{-}svms}$ (g^{dot}) is 2.9, 5.8 and 6.2 points respectively. We note that AMNM$_{bert\text{-}svms}$ (g^{dot}) leads to considerably higher P@{5,10}, metrics that have a pivotal role in practical use of search systems. In addition, PhenoTips shows comparable MAP to BERT but has considerably lower P@{5,10}.[10]

The fusion functions g^{dot} (dot product) and g^{outer} (outer product) outperform the more expensive fusion function g^{conv} (one-dimensional convolution). The lower performance of g^{conv} could be attributed to average pooling, which assumes different input dimensions equally contribute to the final representation and relevance. As a result, it may fail to eliminate noisy features or prioritize important ones.

	Text View			Code View		
Model	**MAP**	**P@5**	**P@10**	**MAP**	**P@5**	**P@10**
BM25	4.1	5.0	3.8	6.5	8.3	6.3
SVMs	8.8	8.3	8.3	7.7	8.3	8.8
BERT	**15.5**	**12.5**	11.7	**10.8**	**13.3**	**10.8**
SVM^{rank}	12.1	9.2	**12.5**	8.5	8.3	8.6

Table 1: MAP, P@5 and P@10 performance of baselines (in percentages) on text and code views.

Model	**Fusion**	**MAP**	**P@5**	**P@10**
SVM^{rank}	text & code	12.9	12.5	12.9
PhenoTips	text & code	15.4	8.3	5.4
AMNM$_{bert\text{-}bert}$	g^{dot}	**18.9***	14.2	17.5
AMNM$_{bert\text{-}bert}$	g^{outer}	18.0*	16.7	17.5
AMNM$_{bert\text{-}bert}$	g^{conv}	16.0*	10.0	12.1
AMNM$_{bert\text{-}svms}$	g^{dot}	18.4*	**18.3**	**17.9**
AMNM$_{bert\text{-}svms}$	g^{outer}	17.1*	17.5	17.1
AMNM$_{bert\text{-}svms}$	g^{conv}	11.4	14.2	13.9

Table 2: Model performance across different fusion functions. The Model column shows the source of representations for text and code views respectively. * indicates significant improvement against best-performing baseline reported in Table 1.

[8] https://trec.nist.gov/trec_eval/

[9] For example, these models can't effectively match a query containing "congestive heart failure" to relevant documents containing "cardiac decompensation," "pulmonary edema," and "ischemic cardiomyopathy."

[10] We note that, in case of rare and undiagnosed diseases, any small improvement is crucial as it can lead to better diagnostic clues. Clinicians often look at the top K results for clues and potential matches for each patient. Therefore, compared to standard evaluation metrics, a more practical evaluation metric for our task is Hit@K, which measures the likelihood of observing "at least one" relevant disease in the ranked list of top K diseases. The Hit@K ($K = 20$) performance of our model is 0.49, while the corresponding value for our best performing baseline is 0.37.

3.2 Model Analysis

We discuss how and why AMNM achieves its improved performance through the following experiments; see supplementary materials for details:

Prediction Variance Across Views: The Pearson correlation between the Average Precision of BERT (text view) and BERT (code view) on individual test queries (patients) is 0.87, which indicates less performance variation across views at query level. This is while the corresponding correlation between BERT (text view) and SVMs (code view) is only 0.34. The lack of diversity in the performance of BERT across these views could be a source of improvement in AMNM$_{bert\text{-}svms}$.

Attention Function: Given test examples (more than 218K patient-disease pairs), our attention sub-network is expected to assign a higher attentive weight to the view that better estimates the corresponding relevance score. To estimate the accuracy of this sub-network, we separately apply the trained BERT (text view) and SVMs (code view) models to generate their corresponding ranked lists of diseases for test patients. Then, for each *relevant* patient-disease pair, we evaluate our attention function in AMNM$_{bert\text{-}svms}$ by measuring whether it assigns a higher attentive weight to the better view–the view that positions the relevant disease at a higher rank compared to the other view. The results show that (a): our attention sub-network is 57.7% accurate in prioritizing better views, (b): BERT (text view) outperforms SVMs (code view) on 64.7% of relevant patient-disease pairs in terms of relative ranks, and our attention network accurately assigns higher weight to BERT on 88.6% of these examples, and (c): on the remaining 35.3% of examples that SVMs (code view) outperforms BERT (text view) in terms of relative ranks, our attention network assigns higher weight to SVMs in only 0.9% of these examples. Improving this percentage could boost the performance of our model and is the subject of our future work.

4 Related Work

The National Institutes of Health established the Undiagnosed Diseases Network (UDN) (Gahl et al., 2015; Ramoni et al., 2017) to facilitate research on undiagnosed and rare diseases. The UDN is a network of 12 clinical sites, and application to the UDN is open to all individuals who complete the application form and submit a referral letter from a health care professional (Kobren et al., 2021). A committee of experts in a review session reviews each UDN application and makes admission decisions. Walley et al. (2018) investigated major factors that may determine application outcomes of the UDN, which has been found effective in developing computational models for predicting admission outcomes (Amiri et al., 2021). In (Dragusin et al., 2013), authors developed a search engine for rare diseases, named FindZebra[11], which was based on information retrieval techniques available in Indri search engine (Strohman et al., 2005). In addition, previous work developed experimental setup to evaluate and compare search engines such as Google or Bing in predicting relevant diseases to given phenotypes (Shenker, 2014), employed medical anthologies and information content techniques (Köhler et al., 2009), leveraged collaborative filtering (Shen et al., 2017) and ensemble techniques (Jia et al., 2018) for this purpose.

Our work departs from previous research by investigating a multiview approach to undiagnosed patients, where we show effective attention and fusion techniques lead to better pairwise ranking for differential diagnosis.

5 Conclusion and Future Work

Given electronic health records of patients, we develop an attentive multiview text representation model to assist clinical experts by ranking the most probable and relevant diseases. Accurate and timely diagnosis is especially important for critically ill patients as it assists specialists to distinguish, prioritize, and accelerate treatment for such patients. Our work can be improved by (a): enriching the feature space through patient- and disease-specific information such patient demographic information and clinical synopsis of diseases, (b): improving model's attention mechanism, and (c): tackling differences in word distributions across patients (queries) and diseases (documents).

Acknowledgments

Research reported in this manuscript was supported by the NIH Common Fund, through the Office of Strategic Coordination/Office of the NIH Director under Award Number U01HG007530. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

[11] https://www.findzebra.com/

Ethics and Broader Impact Statement

This investigation included a small cohort of diagnosed patients in the Undiagnosed Diseases Network (UDN). The UDN is a network of 12 clinical sites, and application to the UDN is open to all individuals who complete the application form and submit a referral letter from a health care professional; a committee of experts in a review session reviews each UDN application and makes admission decisions. We included all data with no exclusions during the data analysis and manual review, except for cases with missing data or formatting issues. The population will therefore reflect the gender, race, ethnicity, age, and health status of the participating patients. In addition, all results have been presented in aggregate and no attempt have been made to identify individuals or facilities. However, during the course of this research and beyond that, there is a potential risk of loss of patient privacy and confidentiality. We have made and will make every effort to protect human subject information and minimize the likelihood of this risk (all authors with access to the data have successfully completed an education program in the protection of human subjects and privacy protection). In addition, our work is transformational in nature and its broader impacts are first and foremost the potential to improve the well-being of individual patients in the society (individuals who often find themselves on a protracted journey from one specialist to another without diagnosis even in this era of genomic sequencing), and support clinicians in their diagnostic efforts.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Joanna S Amberger, Carol A Bocchini, François Schiettecatte, Alan F Scott, and Ada Hamosh. 2015. Omim. org: Online mendelian inheritance in man (omim®), an online catalog of human genes and genetic disorders. *Nucleic acids research*, 43(D1):D789–D798.

Hadi Amiri, Isaac S Kohane, et al. 2021. Machine learning of patient characteristics to predict admission outcomes in the undiagnosed diseases network. *JAMA network open*, 4(2):e2036220–e2036220.

Hadi Amiri, Philip Resnik, Jordan Boyd-Graber, and Hal Daumé III. 2016. Learning text pair similarity with context-sensitive autoencoders. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1882–1892.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Anita Burgun and Olivier Bodenreider. 2001. Comparing terms, concepts and semantic classes in wordnet and the unified medical language system. In *Proceedings of the NAACL'2001 Workshop,"WordNet and Other Lexical Resources: Applications, Extensions and Customizations.*

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard S Dick, Elaine B Steen, Don E Detmer, et al. 1997. *The computer-based patient record: an essential technology for health care*. National Academies Press.

Radu Dragusin, Paula Petcu, Christina Lioma, Birger Larsen, Henrik L Jørgensen, Ingemar J Cox, Lars Kai Hansen, Peter Ingwersen, and Ole Winther. 2013. Findzebra: a search engine for rare diseases. *International Journal of Medical Informatics*, 82(6):528–538.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 16–23. Association for Computational Linguistics.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Carol Friedman, Thomas C Rindflesch, and Milton Corn. 2013. Natural language processing: state of the art and prospects for significant progress, a workshop sponsored by the national library of medicine. *Journal of biomedical informatics*, 46(5):765–773.

William A Gahl, Anastasia L Wise, and Euan A Ashley. 2015. The undiagnosed diseases network of the national institutes of health: a national extension. *Jama*, 314(17):1797–1798.

Marta Girdea, Sergiu Dumitriu, Marc Fiume, Sarah Bowdin, Kym M Boycott, Sébastien Chénier, David Chitayat, Hanna Faghfoury, M Stephen Meyn, Peter N Ray, et al. 2013. Phenotips: Patient phenotyping software for clinical and research use. *Human mutation*, 34(8):1057–1065.

Christine Howes, Matthew Purver, Rose McCabe, Patrick GT Healey, and Mary Lavelle. 2012. Predicting adherence to treatment for schizophrenia from dialogue transcripts. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 79–83. Association for Computational Linguistics.

Jinmeng Jia, Ruiyuan Wang, Zhongxin An, Yongli Guo, Xi Ni, and Tieliu Shi. 2018. Rdad: a machine learning system to support phenotype-based rare disease diagnosis. *Frontiers in genetics*, 9:587.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Shilpa Nadimpalli Kobren, Dustin Baldridge, Matt Velinder, Joel B Krier, Kimberly LeBlanc, Cecilia Esteves, Barbara N Pusey, Stephan Züchner, Elizabeth Blue, Hane Lee, et al. 2021. Commonalities across computational workflows for uncovering explanatory variants in undiagnosed cases. *Genetics in Medicine*, pages 1–11.

Yllka Kodra, Bernardino Fantini, and Domenica Taruscio. 2012. Classification and codification of rare diseases. *Journal of clinical epidemiology*, 65(9):1026–1027.

Sebastian Köhler, Marcel H Schulz, Peter Krawitz, Sebastian Bauer, Sandra Dölken, Claus E Ott, Christine Mundlos, Denise Horn, Stefan Mundlos, and Peter N Robinson. 2009. Clinical diagnostics in human genetics with semantic similarity searches in ontologies. *The American Journal of Human Genetics*, 85(4):457–464.

Sebastian Köhler, Nicole A Vasilevsky, Mark Engelstad, Erin Foster, Julie McMurry, Ségolène Aymé, Gareth Baynam, Susan M Bello, Cornelius F Boerkoel, Kym M Boycott, et al. 2017. The human phenotype ontology in 2017. *Nucleic acids research*, 45(D1):D865–D876.

C Lindberg. 1990. The unified medical language system (umls) of the national library of medicine. *Journal (American Medical Record Association)*, 61(5):40–42.

Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of biomedical informatics*, 40(3):288–299.

Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. 2019. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358.

Rachel B Ramoni, John J Mulvihill, David R Adams, Patrick Allard, Euan A Ashley, Jonathan A Bernstein, William A Gahl, Rizwan Hamid, Joseph Loscalzo, Alexa T McCray, et al. 2017. The undiagnosed diseases network: accelerating discovery about health and disease. *The American Journal of Human Genetics*, 100(2):185–192.

Markus Reuber, Chiara Monzoni, Basil Sharrack, and Leendert Plug. 2009. Using interactional and linguistic analysis to distinguish between epileptic and psychogenic nonepileptic seizures: a prospective, blinded multirater study. *Epilepsy & Behavior*, 16(1):139–144.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 232–241. Springer.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Cícero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 694–699, Beijing, China. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 373–382.

Feichen Shen, Sijia Liu, Yanshan Wang, Liwei Wang, Naveed Afzal, and Hongfang Liu. 2017. Leveraging collaborative filtering to accelerate rare disease diagnosis. In *AMIA Annual Symposium Proceedings*, volume 2017, page 1554. American Medical Informatics Association.

Bennett S Shenker. 2014. The accuracy of internet search engines to predict diagnoses from symptoms can be assessed with a validated scoring system. *International journal of medical informatics*, 83(2):131–139.

Luca Soldaini and Nazli Goharian. 2016. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR Workshop, the 39th international ACM SIGIR conference on research and development in information retrieval*.

Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the international conference on intelligent analysis*, volume 2, pages 2–6. Citeseer.

Nicole M Walley, Loren DM Pena, Stephen R Hooper, Heidi Cope, Yong-Hui Jiang, Allyn McConkie-Rosell, Camilla Sanders, Kelly Schoch, Rebecca C Spillmann, Kimberly Strong, et al. 2018. Characteristics of undiagnosed diseases network applicants: implications for referring providers. *BMC health services research*, 18(1):1–8.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *Deep Learning Workshop, Neural Information Processing*.

MedNLI Is Not Immune: Natural Language Inference Artifacts in the Clinical Domain

Christine Herlihy
Department of Computer Science
University of Maryland
College Park, MD
cherlihy@cs.umd.edu

Rachel Rudinger
Department of Computer Science
University of Maryland
College Park, MD
rudinger@umd.edu

Abstract

Crowdworker-constructed natural language inference (NLI) datasets have been found to contain statistical artifacts associated with the annotation process that allow hypothesis-only classifiers to achieve better-than-random performance (Poliak et al., 2018; Gururangan et al., 2018; Tsuchiya, 2018). We investigate whether MedNLI, a physician-annotated dataset with premises extracted from clinical notes, contains such artifacts (Romanov and Shivade, 2018).

We find that entailed hypotheses contain generic versions of specific concepts in the premise, as well as modifiers related to responsiveness, duration, and probability. Neutral hypotheses feature conditions and behaviors that co-occur with, or cause, the condition(s) in the premise. Contradiction hypotheses feature explicit negation of the premise and implicit negation via assertion of good health. Adversarial filtering demonstrates that performance degrades when evaluated on the *difficult* subset. We provide partition information and recommendations for alternative dataset construction strategies for knowledge-intensive domains.

1 Introduction

In the clinical domain, the ability to conduct natural language inference (NLI) on unstructured, domain-specific texts such as patient notes, pathology reports, and scientific papers, plays a critical role in the development of predictive models and clinical decision support (CDS) systems.

Considerable progress in domain-agnostic NLI has been facilitated by the development of large-scale, crowdworker-constructed datasets, including the Stanford Natural Language Inference corpus (SNLI), and the Multi-Genre Natural Language Inference (MultiNLI) corpus (Bowman et al., 2015; Williams et al., 2017). MedNLI is a similarly-motivated, healthcare-specific dataset created by a small team of physician-annotators in lieu of crowdworkers, due to the extensive domain expertise required (Romanov and Shivade, 2018).

Poliak et al. (2018), Gururangan et al. (2018), Tsuchiya (2018), and McCoy et al. (2019) empirically demonstrate that SNLI and MultiNLI contain lexical and syntactic annotation artifacts that are disproportionately associated with specific classes, allowing a hypothesis-only classifier to significantly outperform a majority-class baseline model. The presence of such artifacts is hypothesized to be partially attributable to the priming effect of the example hypotheses provided to crowdworkers at annotation-time. Romanov and Shivade (2018) note that a hypothesis-only baseline is able to outperform a majority class baseline in MedNLI, but they do not identify specific artifacts.

We confirm the presence of annotation artifacts in MedNLI and proceed to identify their lexical and semantic characteristics. We then conduct adversarial filtering to partition MedNLI into *easy* and *difficult* subsets (Sakaguchi et al., 2020). We find that performance of off-the-shelf `fastText`-based hypothesis-only and hypothesis-plus-premise classifiers is lower on the *difficult* subset than on the *full* and *easy* subsets (Joulin et al., 2016). We provide partition information for downstream use, and conclude by advocating alternative dataset construction strategies for knowledge-intensive domains.[1]

2 The MedNLI Dataset

MedNLI is domain-specific evaluation dataset inspired by general-purpose NLI datasets, including SNLI and MultiNLI (Romanov and Shivade, 2018; Bowman et al., 2015; Williams et al., 2017). Much like its predecessors, MedNLI consists of premise-hypothesis pairs, in which the premises are drawn

[1] See https://github.com/crherlihy/clinical_nli_artifacts for code and partition ids.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1020–1027
August 1–6, 2021. ©2021 Association for Computational Linguistics

from the `Past Medical History` sections of a randomly selected subset of de-identified clinical notes contained in MIMIC-III (Johnson et al., 2016; Goldberger et al., 2000 (June 13). MIMIC-III was created from the records of adult and neonatal intensive care unit (ICU) patients. As such, complex and clinically severe cases are disproportionately represented, relative to their frequency of occurrence in the general population.

Physician-annotators were asked to write a *definitely true*, *maybe true*, and *definitely false* set of hypotheses for each premise, corresponding to *entailment*, *neutral* and *contradiction* labels, respectively. The resulting dataset has cardinality: $n_{\text{train}} = 11232$; $n_{\text{dev}} = 1395$; $n_{\text{test}} = 1422$.

3 MedNLI Contains Artifacts

To determine whether MedNLI contains annotation artifacts that may artificially inflate the performance of models trained on this dataset, we train a simple, premise-unaware, `fastText` classifier to predict the label of each premise-hypothesis pair, and compare the performance of this classifier to a majority-class baseline, in which all training examples are mapped to the most commonly occurring class label (Joulin et al., 2016; Poliak et al., 2018; Gururangan et al., 2018). Note that since annotators were asked to create an entailed, contradictory, and neutral hypothesis for each premise, MedNLI is class-balanced. Thus, in this setting, a majority class baseline is equivalent to choosing a label uniformly at random for each training example.

The micro F1-score achieved by the `fastText` classifier significantly exceeds that of the majority class baseline, confirming the findings of Romanov and Shivade (2018), who report a micro-F1 score of 61.9 but do not identify or analyze artifacts:

	dev	test
majority class	33.3	33.3
`fastText`	**64.8**	**62.6**

Table 1: Performance (micro F1-score) of the `fastText` hypothesis-only classifier.

As the confusion matrix for the test set shown in Table 2 indicates, the `fastText` model is most likely to misclassify entailment as neutral, and neutral and contradiction as entailment. Per-class precision and recall on the test set are highest for contradiction (73.2; 72.8) and lowest for entailment (56.7; 53.8).

	entailment	neutral	contradiction
entailment	**255**	151	68
neutral	126	**290**	58
contradiction	69	60	**345**

Table 2: Confusion matrix for `fastText` classifier.

4 Characteristics of Clinical Artifacts

In this section, we conduct class-specific lexical analysis to identify the clinical and domain-agnostic characteristics of annotation artifacts associated with each set of hypotheses in MedNLI.

4.1 Preprocessing

We cast each hypothesis string in the MedNLI training dataset to lowercase. We then use a `scispaCy` model pre-trained on the `en_core_sci_lg` corpus for tokenization and clinical named entity recognition (CNER) (Neumann et al., 2019a). One challenge associated with clinical text, and scientific text more generally, is that semantically meaningful entities often consist of spans rather than single tokens. To mitigate this issue during lexical analysis, we map each multi-token entity to a single-token representation, where sub-tokens are separated by underscores.

4.2 Lexical Artifacts

Following Gururangan et al. (2018), to identify tokens that occur disproportionately in hypotheses associated with a specific class, we compute token-class pointwise mutual information (PMI) with add-50 smoothing applied to raw counts, and a filter to exclude tokens appearing less than five times in the overall training dataset. Table 3 reports the top 15 tokens for each class.

$$\texttt{PMI}(\text{token}, \text{class}) = log_2 \frac{p(\text{token}, \text{class})}{p(\text{token}, \cdot)p(\cdot, \text{class})}$$

Entailment Entailment hypotheses are characterized by tokens about: (1) patient status and response to treatment (e.g., *responsive*; *failed*; *longer* as in *no longer intubated*); (2) medications and procedures which are common among ICU patients (e.g., *broad_spectrum*; *antibiotics*; *pressors*; *steroid_medication*; *underwent*; *removal*); (3) generalized versions of specific words in the premise (e.g., *comorbidities*; *multiple_medical_problems*), which Gururangan et al. (2018) also observe in SNLI; and (4) modifiers related to duration, frequency, or probability (e.g., *frequent*, *possible*, *high_risk*).

entailment	%	neutral	%	contradiction	%
just	0.25%	cardiogenic_shock	0.33%	no_history_of_cancer	0.27%
high_risk	0.26%	pelvic_pain	0.30%	no_treatment	0.27%
pressors	0.25%	joint_pain	0.30%	normal_breathing	0.27%
possible	0.26%	brain_injury	0.32%	no_history_of_falls	0.27%
elevated_blood_pressure	0.26%	delerium	0.30%	normal_heart_rhythm	0.28%
responsive	0.25%	intracranial_pressure	0.30%	health	0.26%
comorbidities	0.26%	smoking	0.42%	normal_head_ct	0.26%
spectrum	0.27%	obesity	0.41%	normal_vision	0.26%
steroid_medication	0.25%	tia	0.32%	normal_aortic_valve	0.27%
longer	0.26%	acquired	0.31%	bradycardic	0.26%
history_of_cancer	0.26%	head_injury	0.31%	normal_blood_sugars	0.27%
broad	0.26%	twins	0.30%	normal_creatinine	0.28%
frequent	0.25%	fertility	0.30%	cancer_history	0.26%
failed	0.26%	statin	0.30%	cardiac	0.33%
medical	0.29%	acute_stroke	0.30%	normal_chest	0.28%

Table 3: Top 15 tokens by `PMI`(token, class); % of *class* training examples that contain the token.

Neutral Neutral hypotheses feature tokens related to: (1) chronic and acute clinical conditions (e.g., *obesity*; *joint_pain*; *brain_injury*); (2) clinically relevant behaviors (e.g., *smoking*; *alcoholic*; *drug_overdose*); and (3) gender and reproductive status (e.g., *fertility*; *pre_menopausal*). Notably, the most discriminative conditions tend to be commonly occurring within the general population and generically stated, rather than rare and specific. This presumably contributes to the relative difficulty that the hypothesis-only `fastText` model has distinguishing between the entailment and neutral classes.

Contradiction Contradiction hypotheses are characterized by tokens that convey normalcy and good health. Lexically, such sentiment manifests as: (1) explicit negation of clinical severity, medical history, or in-patient status (e.g., *denies_pain*; *no_treatment*; *discharged_home*), or (2) affirmation of clinically unremarkable findings (e.g., *normal_heart_rhythm*; *normal_blood_sugars*), which would generally be rare among ICU patients. This suggests a heuristic of inserting negation token(s) to contradict the premise, which Gururangan et al. (2018) also observe in SNLI.

4.3 Syntactic Artifacts

Hypothesis Length In contrast to Gururangan et al. (2018)'s finding that entailed hypotheses in SNLI tend to be shorter while neutral hypotheses tend to be longer, hypothesis sentence length does not appear to play a discriminatory role in MedNLI, regardless of whether we consider merged- or separated-token representations of multi-word entities, as illustrated by Table 4:

	entailment		**neutral**		**contradiction**	
	mean	median	mean	median	mean	median
separate	5.6	5.0	5.2	5.0	5.6	5.0
merged	5.3	5.0	4.9	5.0	5.3	5.0

Table 4: Average and median hypothesis length by class and entity representation.

5 Physician-Annotator Heuristics

In this section, we re-introduce premises to our analysis to evaluate a set of hypotheses regarding latent, class-specific annotator heuristics. If annotators *do* employ class-specific heuristics, we should expect the semantic contents, φ, of a given hypothesis, $h \in \mathcal{H}$, to be influenced not only by the semantic contents of its associated premise, $p \in \mathcal{P}$, but also by the target class, $c \in \mathcal{C}$.

To investigate, we identify a set of heuristics parameterized by $\varphi(p)$ and c, and characterized by the presence of a set of heuristic-specific Medical Subject Headings (MeSH) linked entities in the premise and hypothesis of each heuristic-satisfying example. These heuristics are described below; specific MeSH features are detailed in the Appendix.

Hypernym Heuristic This heuristic applies when the premise contains clinical condition(s), medication(s), finding(s), procedure(s) or event(s), the target class is *entailment*, and the generated hypothesis contains term(s) that can be interpreted as super-types for a subset of elements in the premise (e.g., clindamycin <: antibiotic).

Probable Cause Heuristic This heuristic applies when the premise contains clinical condition(s), the target class is *neutral*, and the generated hypothesis provides a plausible, often subjective

or behavioral, causal explanation for the condition, finding, or event described in the premise (e.g., associating altered mental status with drug overdose).

Everything Is Fine Heuristic This heuristic applies when the premise contains condition(s) or finding(s), the target class is *contradiction*, and the generated hypothesis negates the premise or asserts unremarkable finding(s). This can take two forms: repetition of premise content plus negation, or inclusion of modifiers that convey good health.

Analysis We conduct a χ^2 test for each heuristic to determine whether we are able to reject the null hypothesis that pattern-satisfying premise-hypothesis pairs are uniformly distributed over classes.

heuristic	χ^2	p-value	top class
hypernym	59.15	1.4e-13‡	entail. (45.2%)
probable cause	111.05	7.7e-25‡	neutral (57.8%)
everything fine	874.71	1.1e-190‡	contradict. (83.8%)

Table 5: Results of χ^2 test statistic by heuristic, computed using the combined MedNLI dataset (‡ $p < 0.001$, † $p < 0.01$, * $p < 0.5$). Top class presented with % of heuristic-satisfying pairs.

The results support our hypotheses regarding each of the three heuristics. Notably, the percentage of heuristic-satisfying pairs accounted for by the top class is lowest for the HYPERNYM hypothesis, which we attribute to the high degree of semantic overlap between entailed and neutral hypotheses.

6 Adversarial Filtering

To mitigate the effect of clinical annotation artifacts, we employ `AFLite`, an adversarial filtering algorithm introduced by Sakaguchi et al. (2020) and analyzed by Bras et al. (2020), to create *easy* and *difficult* partitions of MedNLI.

`AFLite` requires distributed representations of the full dataset as input, and proceeds in an iterative fashion. At each iteration, an ensemble of n linear classifiers are trained and evaluated on different random subsets of the data. A score is then computed for each premise-hypothesis instance, reflecting the number of times the instance is correctly labeled by a classifier, divided by the number of times the instance appears in any classifier's evaluation set. The top-k instances with scores above a threshold, τ, are filtered out and added to the *easy* partition; the remaining instances are retained. This process continues until the size of the filtered subset is $< k$, or the number of retained instances is $< m$; retained instances constitute the *difficult* partition.

To represent the full dataset, we use $\texttt{fastText}_{\text{MIMIC-III}}$ embeddings, which have been pretrained on deidentified patient notes from MIMIC-III (Romanov and Shivade, 2018; Johnson et al., 2016). We represent each example as the average of its component token vectors. We proportionally adjust a subset of the hyperparameters used by Sakaguchi et al. (2020) to account for the fact that MedNLI contains far fewer examples than WINOGRANDE[2]: specifically, we set the training size for each ensemble, m, to 5620, which represents $\approx \frac{2}{5}$ of the MedNLI combined dataset. The remaining hyperparameters are unchanged: the ensemble consists of $n = 64$ logistic regression models, the filtering cutoff, k = 500, and the filtering threshold $\tau = 0.75$.

We apply `AFLite` to two different versions of MedNLI: (1) $\mathcal{X}_{h,m}$: hypothesis-only, multi-token entities merged, and (2) $\mathcal{X}_{ph,m}$: premise and hypothesis concatenated, multi-token entities merged. `AFLIte` maps each version to an *easy* and *difficult* partition, which can in turn be split into training, dev, and test subsets. We report results for the `fastText` classifier trained on the original, hypothesis-only (hypothesis + premise) MedNLI training set, and evaluated on the *full*, *easy* and *difficult* dev and test subsets of $\mathcal{X}_{h,m}$ ($\mathcal{X}_{ph,m}$), and observe that performance decreases on the *difficult* partition:

	model	eval dataset	full	easy (Δ)	difficult (Δ)
no premise	majority class	dev	0.33	0.34 (+0.01)	0.35 (+0.02)
no premise	majority class	test	0.33	0.35 (+0.02)	0.37 (+0.04)
no premise	`fastText`	dev	0.65	0.67 (+0.02)	0.46 (-0.19)
no premise	`fastText`	test	0.63	0.65 (+0.02)	0.4 (-0.23)
with premise	majority class	dev	0.33	0.45 (+0.12)	0.36 (+0.03)
with premise	majority class	test	0.33	0.48 (+0.15)	0.37 (+0.04)
with premise	`fastText`	dev	0.53	0.6 (+0.07)	0.43 (-0.1)
with premise	`fastText`	test	0.51	0.55 (+0.04)	0.4 (-0.11)

Table 6: Performance (micro F1-score) for the majority class baseline and `fastText` classifiers, with and without premise, by partition (e.g., *full, easy, difficult*).

7 Discussion

7.1 MedNLI is Not Immune from Artifacts

In this paper, we demonstrate that MedNLI suffers from the same challenge associated with annotation artifacts that its domain-agnostic predecessors have

[2]MedNLI's training dataset contains 14049 examples when the training, dev, and test sets are combined, while WINOGRANDE contains 47K after excluding the 6K used for fine-tuning.

encountered: namely, NLI models trained on {Med, S, Multi}NLI can perform well even without access to the training examples' premises, indicating that they often exploit shallow heuristics, with negative implications for out-of-sample generalization.

Interestingly, many of the high-level lexical characteristics identified in MedNLI can be considered domain-specific variants of the more generic, class-specific patterns identified in SNLI. This observation suggests that a set of abstract design patterns for inference example generation exists across domains, and may be reinforced by the prompts provided to annotators. Creative or randomized priming, such as Sakaguchi et al. (2020) 's use of anchor words from WikiHow articles, may help to decrease reliance on such design patterns, but it appears unlikely that they can be systematically sidestepped without introducing new, "corrective" artifacts.

7.2 A Prescription for Dataset Construction

To mitigate the risk of performance overestimation associated with annotation artifacts, Zellers et al. (2019) advocate adversarial dataset construction, such that benchmarks will co-evolve with language models. This may be difficult to scale in knowledge-intensive domains, as expert validation of adversarially generated benchmarks is typically required. Additionally, in high-stakes domains such as medicine, information-rich inferences should be preferred over correct but trivial inferences that time-constrained expert annotators may be rationally incentivized to produce, because entropy-reducing inferences are more useful for downstream tasks.

We advocate the adoption of a mechanism design perspective, so as to develop modified annotation tasks that reduce the cognitive load placed on expert annotators while incentivizing the production of domain-specific NLI datasets with high downstream utility (Ho et al., 2015; Liu and Chen, 2017). An additional option is to narrow the generative scope by defining a set of inferences deemed to be useful for a specific task. Annotators can then map (premise, relation) tuples to relation-satisfying, potentially fuzzy subsets of this pool of useful inferences, or return partial functions when more information is needed.

8 Ethical Considerations

When working with clinical data, two key ethical objectives include: (1) the preservation of patient privacy, and (2) the development of language and predictive models that benefit patients and providers to the extent possible, without causing undue harm. With respect to the former, MedNLI's premises are sampled from de-identified clinical notes contained in MIMIC-III (Goldberger et al., 2000 (June 13; Johnson et al., 2016), and the hypotheses generated by annotators do not refer to specific patients, providers, or locations by name. MedNLI requires users to complete Health Insurance Portability and Accountability Act (HIPAA) training and sign a data use agreement prior to being granted access, which we have complied with.

Per MedNLI's data use agreement requirements, we do not attempt to identify any patient, provider, or institution mentioned in the de-identified corpus. Additionally, while we provide AFLite *easy* and *difficult* partition information for community use in the form of split-example ids and a checksum, we do not share the premise or hypothesis text associated with any example. Interested readers are encouraged to complete the necessary training and obtain credentials so that they can access the complete dataset (Romanov and Shivade, 2018; Goldberger et al., 2000 (June 13).

With respect to benefiting patients, the discussion of natural language artifacts we have presented is intended to encourage clinical researchers who rely on (or construct) expert-annotated clinical corpora to train domain-specific language models, or consume such models to perform downstream tasks, to be aware of the presence of annotation artifacts, and adjust their assessments of model performance accordingly. It is our hope that these findings can be used to inform error analysis and improve predictive models that inform patient care.

Acknowledgments

We thank the four anonymous reviewers whose feedback and suggestions helped improve this manuscript. The first author was supported by the National Institute of Standards and Technology's (NIST) Professional Research Experience Program (PREP). This research was also supported by the DARPA KAIROS program. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of NIST, DARPA, or the U.S. Government.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases.

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. 2000 (June 13). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 419–429, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Yang Liu and Yiling Chen. 2017. Machine-learning aided peer prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, page 63–80, New York, NY, USA. Association for Computing Machinery.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019a. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019b. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from Natural Language Inference in the Clinical Domain. *CoRR*, abs/1808.06752.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *CoRR*, abs/1704.05426.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Hypothesis-only Baseline Analysis

To conduct the analysis presented in Section 3, we take the MedNLI training dataset as input, and exclude the premise text for each training example. We cast the text of each training hypothesis to lowercase, but do not perform any additional preprocessing. We use an off-the-shelf `fastText` classifier, with all model hyperparameters set to their default values with the exception of `wordNgrams`, which we set equal to 2 to allow the model to use bigrams in addition to unigrams (Joulin et al., 2016). We evaluate the trained classifier on the hypotheses contained in the MedNLI dev and test datasets, and report results for each split.

A.2 Lexical Artifact Analysis

To perform the analysis presented in Section 4, we cast each hypothesis string in the MedNLI training dataset to lowercase. We then use a `scispaCy` model pre-trained on the `en_core_sci_lg` corpus for tokenization and clinical named entity recognition (CNER) (Neumann et al., 2019a). Next, we merge multi-token entities, using underscores as delimiters—e.g., "brain injury" → "brain_injury".

When computing token-class pointwise mutual information (PMI), we exclude tokens that appear less than five times in the overall training dataset's hypotheses. Then, following Gururangan et al. (2018), who apply add-100 smoothing to raw counts to highlight particularly discriminative token-class co-occurrence patterns, we apply add-50 smoothing to raw counts. Our approach is similarly motivated; our choice of 50 reflects the smaller state space associated with a focus on the clinical domain.

A.3 Semantic Analysis of Heuristics

To perform the statistical analysis presented in Section 5, we take the premise-hypothesis pairs from the MedNLI training, dev, and test splits, and combine them to produce a single corpus. We use a `scispaCy` model pre-trained on the `en_core_sci_lg` corpus for tokenization and entity linking (Neumann et al., 2019b), and link against the Medical Subject Headings (MeSH) knowledge base. We take the top-ranked knowledge base entry for each linked entity. Linking against MeSH provides a unique concept id, canonical name, alias(es), a definition, and one or more MeSH tree numbers for each recovered entity. Tree numbers convey semantic type information by embedding each concept into the broader MeSH hierarchy [3]. We operationalize each of our heuristics with a set of MeSH-informed semantic properties, which are defined as follows:

1. **Hypernym Heuristic:** a premise-hypothesis pair satisfies this heuristic if specific clinical concept(s) appearing in the premise appear in a more general form in the hypothesis. Formally: $\{(p,h)|\varphi(p) \subsetneq \varphi(h)\}$. MeSH tree numbers are organized hierarchically, and increase in length with specificity. Thus, when a premise entity and hypothesis entity are left-aligned, the hypothesis entity is a hypernym for the premise entity if the hypothesis entity is a substring of the premise entity. To provide a concrete example: *diabetes mellitus* is an *endocrine system disease*; the associated MeSH tree numbers are C19.246 and C19, respectively.

2. **Probable Cause Heuristic:** a premise-hypothesis pair satisfies this heuristic if: (1) the premise contains one or more MeSH entities belonging to high-level categories C (diseases), D (chemicals and drugs), E (analytical, diagnostic and therapeutic techniques, and equipment) or F (psychiatry and psychology); and (2) the hypothesis contains one or more MeSH entities that can be interpreted as providing a plausible causal or behavioral explanation for the condition, finding, or event described in the premise (e.g., smoking, substance-related disorders, mental disorders, alcoholism, homelessness, obesity).

3. **Everything Is Fine Heuristic:** a premise-hypothesis pair satisfies this heuristic if the hypothesis contains one or more of the same MeSH entities as the premise (excluding the *patient* entity, which appears in almost all notes) and also contains: (1) a negation word or phrase (e.g., *does not have*, *no finding*, *no*, *denies*); or (2) a word or phrase that affirms the patient's health (e.g., *normal*, *healthy*, *discharged*).

For each heuristic, we subset the complete dataset to find pattern-satisfying premise-heuristic pairs. We use this subset when performing the χ^2 tests.

[3] https://meshb.nlm.nih.gov/treeView

A.4 Adversarial Filtering

When implementing `AFLite`, we follow Sakaguchi et al. (2020). We use a smaller training set size of $m = 5620$, but keep the remaining hyperparameters unchanged, such that the ensemble consists of $n = 64$ logistic regression models, the filtering cutoff, $k = 500$, and the filtering threshold $\tau = 0.75$.

Towards a more Robust Evaluation for Conversational Question Answering

Wissam Siblini, Baris Sayil, Yacine Kessaci
Worldline, France
{wissam.siblini,yacine.kessaci}@worldline.com
baris.sayil@insa-lyon.fr

Abstract

With the explosion of chatbot applications, Conversational Question Answering (CQA) has generated a lot of interest in recent years. Among proposals, reading comprehension models which take advantage of the conversation history (previous QA) seem to answer better than those which only consider the current question. Nevertheless, we note that the CQA evaluation protocol has a major limitation. In particular, models are allowed, at each turn of the conversation, to access the ground truth answers of the previous turns. Not only does this severely prevent their applications in fully autonomous chatbots, it also leads to unsuspected biases in their behavior. In this paper, we highlight this effect and propose new tools for evaluation and training in order to guard against the noted issues. The new results that we bring come to reinforce methods of the current state of the art.

1 Introduction

The ability to automatically answer questions from a set of raw text paragraphs has long been coveted by computer scientists (Woods, 1977). For applications in search engines, one could consider an isolated task where a user formulates a single question (Croft et al., 2010; Siblini et al., 2020). But recently, with usage in conversational agents (e.g. chatbots), a more contextualized variant referred to as Conversational Question Answering (CQA) has attracted a great deal of attention (Reddy et al., 2019; Choi et al., 2018). CQA differs from traditional (extractive) Question Answering (Rajpurkar et al., 2016) because Question-Answer (QA) pairs are not single but come in sequences within conversations. Therefore, models can use previous turns as context to extract the answer of the current question (Zhu et al., 2018; Huang et al., 2018; Qu et al., 2019a). In some cases, the history is even crucial to disambiguate pronouns in the question.

Similarly to other NLP tasks, the state-of-the-art approaches for CQA are variants of the Transformer Encoder (Vaswani et al., 2017), a deep neural network with several self-attention layers that produce contextualized representations of the "tokens" (words, subwords) that compose a text. For instance, models like BERT (Devlin et al., 2019; Lan et al., 2019; Sanh et al., 2019) obtain a more than decent performance on CQA datasets like QuAC (Choi et al., 2018) or CoQA (Reddy et al., 2019). However, they miss the context to fully understand the questions. Proposals have been made to integrate the history in several manners: using a recursive strategy (Huang et al., 2018), appending previous QAs to the current question as input (Zhu et al., 2018), and contextualizing the question-paragraph pair with respect to the history. We can mention in particular BERT-HAE and BERT-PHAE (Qu et al., 2019a,b) which improve BERT in a simple yet efficient way by encoding, in addition to segment and position, the fact that parts of the paragraph's words belonged to previous answers.

2 Motivation and main contributions

Our objective here is not to propose yet another model to try to obtain the best predictive score on CQA leaderboards. Instead, we focus our thinking around the current evaluation/training protocols with regards to the possible application cases. The starting point of our reflection is that currently, when evaluated on CQA datasets, models like BERT-HAE use the ground-truth answers of previous turns as context to answer the current question. This limits the scope of applicability to only a "semi-automatic" bot that would require a human providing supervision at each turn. We also show how it biases the selection of models towards those with an undesired filter behavior.

To make approaches from the literature usable

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1028–1034
August 1–6, 2021. ©2021 Association for Computational Linguistics

in more difficult/realistic scenarios like standalone chatbots (in which they can only access the previous questions and their predictions of the answers), we make the following contributions: **(1)** We implement new evaluation tools to first highlight the current unnoticed and undesirable behavior: in ground-truth free conditions, CQA approaches can become even less accurate than baselines like BERT which do not exploit the history at all. **(2)** We develop the analog training protocol to make approaches robust to the observed issues. In particular, this gives back state-of-the-art models the strength to outperform the baseline but this time in a scenario that connects better to real-world conversational agents. Our work comes with an implementation of conversational QA tools, based on the most widely used transformers library (Wolf et al., 2019).

3 Conversational Question Answering

Conversational Question Answering (CQA) is a Natural Language Processing task related to Machine Comprehension (MC) (Zhang et al., 2019; Gupta et al., 2020). MC has grown significantly over the last decade, particularly thanks to (1) large scale datasets such as SQuAD (Rajpurkar et al., 2016) or Natural Questions (Kwiatkowski et al., 2019), (2) the improvement of representation learning models (Joulin et al., 2017), (3) powerful mechanisms such as attention (Yang et al., 2016; Vaswani et al., 2017), and (4) the emergence of several related topics like multi-lingual modeling (Pires et al., 2019; Siblini et al., 2019) or Conversational Question Answering (Choi et al., 2018; Reddy et al., 2019).

In CQA, questions are grouped in conversations and often require the context, i.e. previous QA turns, to be fully understandable. QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2019) are two examples of CQA datasets. They were both generated by humans (a "student" and a "teacher") through conversations where the student asks a series of questions, complementary or not, on a given paragraph and the teacher answers them. In this paper, we focus on QuAC (Question Answering in Context) which is more recent and described as more challenging than CoQA (Choi et al., 2018). It contains 14k conversations and around 100k question-paragraph pairs, split into a training set (11,567 conversations / 83,568 questions), a validation set (1,000 conversations / 7,354 questions) and a test set. It evaluates models with several metrics, the main one being the F1-score (Flach, 2003).

Models proposed for QuAC are similar to those developed for SQuAD (e.g. BiDAF (Seo et al., 2016) or BERT (Devlin et al., 2019)) but they additionally integrate the history. A popular example is BERT-HAE (Qu et al., 2019a). It uses BERT's architecture but modifies the input embedding layer to add a novel component: the History Answer Embedding (HAE). As usual, the input question-paragraph pair is tokenized and marked with positions and segments. Then, an additional History Answer marker is added to indicate whether the tokens belonged to answers of previous questions or not, and the resulting embedding is simply added to the other embedding vectors (token, position, segment) before the self-attention blocks. BERT-HAE was enhanced, in a later publication (Qu et al., 2019b) by BERT-PHAE (Positional HAE) which additionally encodes the turn position of the answers in the history. Although very promising, we note that BERT-HAE and BERT-PHAE, as well as other state of the art models for QuAC, access the ground-truth answers of previous turns during evaluation. Therefore, reported results only reflect the performance within a reduced scope of applicability. In the following, we detail this limitation and propose to complement the current protocol in order to improve both evaluation and training.

4 A more Robust Protocol

Consider a standalone chatbot that successively answers questions from documents. At each turn, it cannot know for sure the ground truth (GT) answers of the previous turns except if the user or another human provides supervision. This could happen in scenarios where the role of the algorithm is only to provide answer suggestions (semi-automatic) to a human agent (e.g. in customer support). However, applications often seek bots where the question-answer loop is automated (standalone). Here we investigate this second setting. We start by reproducing the literature results on the semi-automatic scenario, then we exhibit the limits and propose solutions for our target scenario.

4.1 Reproducing the Regular Evaluation in the Semi-automatic Scenario

To evaluate the baseline performance (semi-automatic), we train BERT-HAE and BERT-PHAE on QuAC using the protocol described by the authors (Qu et al., 2019a) and the same hyperparame-

ters: history markers from up to 6 turns, and specific optimization parameters (12 as batch size, 3e-5 as learning rate with a linear decrease to 0 over 24k training steps). We implement our own training script on the basis of codes pieces from the transformers library (Wolf et al., 2019) and BERT-HAE's authors[1]. Experiments are run with a Nvidia Tesla V100 GPU.

Model	F1	Uses history
BiDAF++ (Choi et al., 2018)	51.8	No
BERT (Qu et al., 2019a)	54.4 (54.8)	No
BERT-HAE (Qu et al., 2019a)	63.1 (63.4)	Yes
BERT-PHAE (Qu et al., 2019b)	64.7 (64.4)	Yes

Table 1: F1-score of BERT, BERT-HAE, BERT-PHAE and a previous baseline on QuAC using the regular evaluation protocol. We display the original results published by the authors and the ones we reproduced (in parentheses).

Our results are roughly equal to those previously reported (Table 1). BERT's F1 score is 54.8, which compares favorably to previous baselines such as BiDAF. By adjusting the representation of the tokens based on the history of answers, BERT-HAE allows a significant improvement to 63.4 (+15.7 %). The position of the turns in the history also has its importance allowing BERT-PHAE to further improve the F1-score to 64.4. This is probably because questions are often related to the answers that directly precede them. To improve the results even further, one can also select a specific subset of turns in the history (Qu et al., 2019b).

4.2 Critical Analysis: The Filtering Behavior

Although promising, the aforementioned results need to be considered with caution. A hasty conclusion is that adding the history allows the model to benefit from a context and hence to better process the current question. However the improvement could also be explained by a bias in the dataset at hand. Indeed, this question answering task is extractive, i.e. answers are selected from a paragraph. In the course of a conversation in QuAC, an average of 7 questions are successively asked on the same rather small paragraph. Thus simply filtering the paragraph tokens with the answer history provides the advantage of reducing considerably the list of possible remaining answers. Note however that such a filtering could also have a negative effect, in the presence of overlap between answers.

[1] https://github.com/prdwb/bert_hae

To get better insights of the impact of a filtering behavior in practice, we run three experiments.

Model	F1	F1 w/ post filtering
BEST	**95.6**	92.7
BERT	54.8	**56.9**
BERT-HAE	**63.4**	62.5

Table 2: Evaluation of the impact of post-filtering on BEST, BERT and BERT-HAE.

Experiment 1: The negative impact of filtering due to overlap We first compute the best reachable F1-score (that we refer to as BEST) as if we had a model that always predicts the expected answer. Then we compute "BEST w/ post filtering" with the same predictions except that we post-filter all tokens that belong the 6 previous turns' answers, except for the "Cannot Answer" tokens (reserved for unanswerable questions). BEST F1 score is 95.6[2] while "BEST F1 w/ post filtering" is lower but very close: 92.7 (Table 2). This tells us that the maximal negative impact of a filtering strategy on QuAC is weak. We find an explanation by doing proportion measurements in QuAC's eval set: in particular, the percentage of overlapping tokens (resp. non overlapping tokens) between answers is low (resp. high): 5.7% (resp. 74.1%), the other 20.2% being the "Cannot Answer" tokens.

Experiment 2: Global impact of a post filtering on the models After 6 turns, sometimes almost half of the paragraph tokens belong to the history of answers. Even if experiment 1 suggests a negative impact of filtering due to overlap, the positive impact on our baselines (due to the significant reduction of the number of candidate answers) could counterbalance. We therefore re-evaluate the models trained in section 4.1, but this time we apply a post processing of their predictions: the start/end logits of tokens that belong to the answers of previous turns are set to $-\infty$, except for the "Cannot Answer" tokens. This forces previous answers to be excluded from the final predicted span text. This simple strategy to integrate the history in BERT allows an improvement to an F1-score of 56.9 (Table 2). On the contrary, it globally reduces the score

[2] Intuitively, it should be 100. But this value is unreachable in practice. The reason is that questions in QuAC have several acceptable answers (span texts of various length) and we select one randomly as BEST prediction. And, QuAC's official evaluation script computes, for each sample, the average F1 between the prediction and all possible answers.

of BERT-HAE to 62.5 (a reduction factor slightly lower than with BEST). These results suggest that access to ground-truth answers of previous turns allows in QuAC, in which the overlap is weak, a filtering mechanism to be a positive way of integrating history. Results also suggest that BERT-HAE might already implicitly integrate a filtering behavior. Unquestionably, it does it in a more expressive manner than our hard post-processing, since the history markers are passed as inputs to the model.

Experiment 3: Does BERT-HAE exhibit a filter behavior? Although suggested by the previous experiment, we want to answer this question more clearly. We consider an experiment aligned with the philosophy of adversarial attacks (Akhtar and Mian, 2018; Morris et al., 2020). During evaluation, we systematically modify the history answer markers so that the tokens of the current expected answer are marked as if they belonged to the history. The results obtained from this evaluation protocol are displayed under the column "F1 w/ Adv" in Table 3. F1 w/ Adv allows to measure, with the F1 metric, the ability of the models to answer a question when its answer has already appeared in the conversation before. In this condition, we observe a dramatic drop in BERT-HAE's performance (from 63.4 to 41.7), and an even worse for BERT-PHAE. This confirms that these models tend to output lower probabilities for tokens that are in the history, which suggests a filtering behavior and makes their usage potentially counter productive.

4.3 Proposed Evaluation for the Standalone Scenario

The current evaluation protocol on QuAC's validation set can bias model selection towards those able to implement a filtering behavior, which seems to be the case for BERT-(P)HAE. Thus, it does not guarantee a robust behavior in a fully autonomous bot. Here we propose an extension.

Inspired by the literature of recurrent models, we refer to the regular evaluation protocol, which access to ground truth answers of previous turns, as the **"Teacher Forcing" (w/ TF) protocol**. Analogically, we consider a mode **"without Teacher Forcing" (w/o TF)** where models process a conversation in the natural order and only use their predictions as history. The latter is outlined in Algorithm 1, where "build_mark" refers to a function that computes the new HAE markers given the previous ones and the new answer. Note that the algorithm for evaluation w/ TF simply replaces "build_mark(HAE,$\text{answer}_{\text{pred}}$)" with "build_mark(HAE,$\text{answer}_{\text{GT}}$)".

Algorithm 1 Evaluation w/o TF

```
s ← 0
for conversation ∈ valid set do
    HAE ← None
    for turn ∈ conversation do
        question ← turn['question']
        answer_GT ← turn['answer']
        answer_pred ← model(question, HAE)
        HAE ← build_mark(HAE,answer_pred)
        s ← s + F1(answer_pred,answer_GT)
    end for
end for
return s / card(valid set)
```

When we take the models trained in section 4.1 (w/ TF) and evaluate them with the new standalone protocol (w/o TF), Table 3 shows that the performance drops from 63.4 to 53.5 with BERT-HAE and from 64.4 to 54.2 with BERT-PHAE. Concretely, although unsuspected with the original protocol, the approaches do not necessarily seem advantageous compared to BERT here. This in no way detracts the interest of these proposals, which implement clever architectures to integrate the history. It only prevents their application, as is, in the standalone scenario. Nevertheless, now that this issue is identified, we can try to design an appropriate strategy to avoid it from the start, by taking measures at the training phase.

4.4 Training for the Standalone Scenario

To complement the proposed evaluation protocol with a training one, we propose to apply a recipe inspired by the most popular defense mechanism against adversarial attacks called adversarial training (Ren et al., 2020), i.e. we introduce the disruptive element (here the mode without Teacher Forcing) at training time. We consider three heuristics: (1) we disable TF during all the training steps (Robust), (2) we disable TF randomly based on a Coin Flip (Robust-CF), (3) we progressively disable TF from 0% of the steps to 100% of the steps over the training iterations (Robust-P). The new training process is detailed in Algorithm 2, where "update" refers to the optimization algorithm that updates the model based on the loss and "heuristic.condition" is a condition that depends on the

heuristic (e.g. always true for heuristic (1)). Note that both heuristics (2) and (3) are inspired from scheduled sampling methods (Bengio et al., 2015) adapted to the context of CQA.

Algorithm 2 Robust Training

```
for conversation ∈ train set do
    HAE ← None
    for turn ∈ conversation do
        question ← turn['question']
        answer_GT ← turn['answer']
        answer_pred ← model(question, HAE)
        l ← loss(answer_pred,answer_GT)
        update(model,l)
        if heuristic.condition then
            answer_add ← answer_pred
        else
            answer_add ← answer_GT
        end if
        HAE ← build_mark(HAE,answer_add)
    end for
end for
return model
```

We obtain encouraging results (Table 3). In particular, BERT-PHAE Robust-P reaches a F1-score of 58.1 in the standalone scenario which is better than BERT's F1. Besides, "F1 /w Adv" for BERT-(P)HAE Robust seems to indicate that, the less we apply TF, the less the entailed model exhibits a filtering behavior. In fact, all the robust variants exhibit a weaker filtering behaviour than the original methods.

Model	F1 w/ TF	F1 w/o TF	F1 w/ Adv
BERT	-	54.4	-
BERT-HAE	**63.4**	53.5	41.7
BERT-HAE Robust	59.5	56.6	**51.7**
BERT-HAE Robust-CF	61.6	55.9	47.4
BERT-HAE Robust-P	60.7	**56.7**	50.7
BERT-PHAE	**64.4**	54.2	40.7
BERT-PHAE Robust	60.5	57.4	**53.3**
BERT-PHAE Robust-CF	62.2	56.4	47.7
BERT-PHAE Robust-P	62.4	**58.1**	51.6
BERT-AH	-	**58.3**	-

Table 3: Evaluation of BERT, BERT-HAE, BERT-PHAE, BERT-AH and the robust variants with different validation protocols.

Our experiment and results leave room for improvement with additional considerations on protocols/parameters/models. For instance, contrary to answers, standalone models can have access to the exact history of questions. What if we integrated the latter instead of the answer history in the model's input? We tested this by implementing a simple model that we refer to as BERT-AH (Appended History) in which previous questions are added to the regular BERT's inputs, and marked with a special embedding. BERT-AH obtains an F1-score of 58.3 (whatever the evaluation protocol, since answer history is not used). Thus, our guess is that the best direction for standalone CQA lies towards both the integration of previous questions and the robust integration of previous answers.

5 Conclusion

The work presented in this paper comes to complement the current training and evaluation protocols for CQA. It allows (1) highlighting unnoticed and undesirable behavior in existing approaches from the literature and (2) more robustness for their application in autonomous chatbots. We hope that this will encourage additional proposals in the same direction. Several improvements could be made in the future. First, because without Teacher Forcing the history is now predicted and not fixed, we could explore the impact of updating the model by back-propagating an answer's error through all previous turns and not only the current one. This would be analog to backpropagation through time. Second, we could augment the current CQA datasets or propose new ones to prevent the biases we observed: for example, QuAC could have conversations including wrong answers, since this occurs in real-life, so that models could be properly trained for. The associated turns would of course only be used as a part of histories. Finally, we should perform user tests to evaluate the robustness of models in real-life, because when a model's answer is wrong, we expect it to impact the next user's question(s). And this cannot be taken into account with the current protocol since the datasets are static.

References

Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in con-

text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter A Flach. 2003. The geometry of roc space: understanding machine learning metrics through roc isometrics. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 194–201.

Somil Gupta, Bhanu Pratap Singh Rawat, and Hong Yu. 2020. Conversational machine comprehension: a literature review. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2739–2753, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. In *International Conference on Learning Representations*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136.

Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W Bruce Croft, and Mohit Iyyer. 2019b. Attentive history selection for conversational question answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1391–1400.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Wissam Siblini, Mohamed Challal, and Charlotte Pasqual. 2020. Delaying interaction layers in transformer-based encoders for efficient open domain question answering. *arXiv preprint arXiv:2010.08422*.

Wissam Siblini, Charlotte Pasqual, Axel Lavielle, Mohamed Challal, and Cyril Cauchois. 2019. Multilingual question answering from formatted text applied to conversational agents. *arXiv preprint arXiv:1910.04659*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

William A. Woods. 1977. Lunar rocks in natural english: Explorations in natural language question answering. *Linguistic Structures Processing*, pages 521–569.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Xin Zhang, An Yang, Sujian Li, and Yizhong Wang. 2019. Machine reading comprehension: a literature review. *arXiv preprint arXiv:1907.01686*.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *arXiv preprint arXiv:1812.03593*.

VAULT: VAriable Unified Long Text Representation for Machine Reading Comprehension

Haoyang Wen[2†*], **Anthony Ferritto**[1†], **Heng Ji**[2]
Radu Florian[1], **Avirup Sil**[1]
[1] IBM Research AI, [2] University of Illinois Urbana-Champaign
wen17@illinois.edu, aferritto@ibm.com
hengji@illinois.edu, {raduf, avi}@us.ibm.com

Abstract

Existing models on Machine Reading Comprehension (MRC) require complex model architecture for effectively modeling long texts with paragraph representation and classification, thereby making inference computationally inefficient for production use. In this work, we propose VAULT: a light-weight and parallel-efficient paragraph representation for MRC based on contextualized representation from long document input, trained using a new Gaussian distribution-based objective that pays close attention to the partially correct instances that are close to the ground-truth. We validate our VAULT architecture showing experimental results on two benchmark MRC datasets that require long context modeling; one Wikipedia-based (Natural Questions (NQ)) and the other on TechNotes (TechQA). VAULT can achieve comparable performance on NQ with a state-of-the-art (SOTA) complex document modeling approach while being 16 times faster, demonstrating the efficiency of our proposed model. We also demonstrate that our model can also be effectively adapted to a completely different domain – TechQA – with large improvement over a model fine-tuned on a previously published large PLM.

1 Introduction

Machine Reading Comprehension (MRC) has seen great advances in recent years with the rise of pre-trained language models (PLM) (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019) and public leaderboards (Rajpurkar et al., 2016, 2018; Yang et al., 2018; Joshi et al., 2017; Welbl et al., 2018; Kwiatkowski et al., 2019). While some challenges (Rajpurkar et al., 2016, 2018) focus on reading comprehension with shorter contexts, many others (Welbl et al., 2018; Joshi et al., 2017; Kwiatkowski et al., 2019; Tanaka et al., 2021) focus on longer contexts that cannot fit into a typical 512 sub-token transformer window. Motivated by this, we focus on reading comprehension with long contexts.

One newer approach to this task (Zheng et al., 2020) focuses on modeling document hierarchy to represent multi-grained information for answer extraction. Although this approach creates a strong representation of the text, it suffers from a significant drawback. The graph-based methods (Veličković et al., 2018) are inefficient on parallel hardware, such as GPUs, resulting in slow inference speed (Zhou et al., 2018; Zheng et al., 2020). Motivated by this, in this paper, we propose a reading comprehension model that addresses the above issue and uses a more light-weight, parallel-efficient (i.e. efficient on parallel hardware) paragraph representation based on long contextual representations for providing paragraph answers to questions. Instead of modeling document hierarchy from tokens to document pieces, we first introduce a base model that builds on top of a large "long-context" PLM (we use Longformer, Beltagy et al., 2020) to model longer contexts with lightweight representations of each paragraph. We note that while our approach could work with any PLM, we expect it to perform better with models that can support long contexts and therefore see more paragraph representations at once (Gong et al., 2020). To provide our model a notion of paragraph position relative to a text we also introduce position-aware paragraph representations (PAPR) utilizing special markup tokens and provide them as input for efficient paragraph classification. This approach allows us to encode paragraph-level position in the text and teach the model to impute information on each paragraph into the hidden outputs for these tokens that we can exploit to determine in which paragraph the answer resides. We then predict the

[*] Work done during an internship at IBM Research AI.
[†] Equal contributions.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1035–1042
August 1–6, 2021. ©2021 Association for Computational Linguistics

answer span from this identified paragraph.

While previous MRC methods (Chen et al., 2017; Devlin et al., 2019) use ground-truth start and end span positions exclusively as training objectives when extracting answer spans from the context and consider all other positions as incorrect instances equally. However, spans that overlap with the ground-truth should be considered as partially correct. Motivated by Li et al. (2020) which proposes a new optimization criteria based on constructing prior distribution over synonyms for machine translation, we further propose to improve the above base model by considering the start and end positions of ground-truth answer spans as Gaussian-like distributions, instead of single points, and optimize our model using statistical distance.

We call this final model, VAULT (**VA**riable **U**nified **L**ong **T**ext representation) as it can handle a variable number and lengths of paragraphs at any position with the same unified model structure to handle long texts.

To evaluate the performance of VAULT, we select the new Natural Questions (NQ, Kwiatkowski et al., 2019) and TechQA (Castelli et al., 2020) datasets. NQ attempts to make Machine Reading Comprehension (MRC) more realistic by providing longer Wikipedia documents as contexts and real user search-engine queries as questions, and aims at avoiding *observation bias*: high lexical overlap between the question and the answer context which can happen frequently if the question is created after the user sees the paragraph (Rajpurkar et al., 2016, 2018; Yang et al., 2018; Chakravarti et al., 2020; Karpukhin et al., 2020; Lee et al., 2019; Murdock et al., 2018). The task introduces the extraction of long answers (henceforth LA; typically paragraphs) besides also requiring short answers (henceforth SA) similar to SQuAD (Rajpurkar et al., 2016). In Figure 1 we examine an example from NQ along with the answers of VAULT and (Zheng et al., 2020). We see that while VAULT can extract answers from the very bottom of a page – if relevant – the existing system suffers from positional bias. It often predicts answers from the first paragraph of Wikipedia (a region which often contains the most relevant information). We evaluate our model for domain adaptation on TechQA, a recently introduced challenging dataset for QA on technical support articles where answers are typically 3-5 times longer than standard MRC datasets (Rajpurkar et al., 2016, 2018).

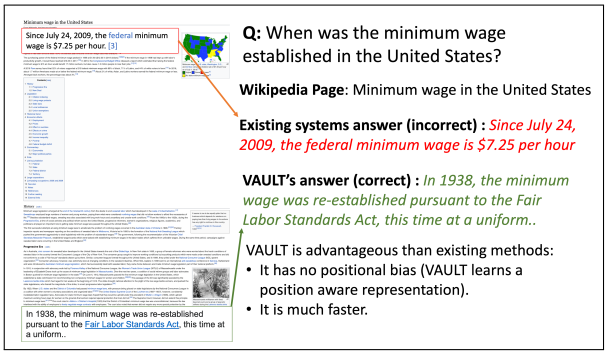


Figure 1: Example from the NQ dataset with answers from VAULT and (Zheng et al., 2020).

Empirically we first show that VAULT achieves comparable performance on NQ with (Zheng et al., 2020)'s document modeling architecture based on graph neural networks while being 16 times faster, demonstrating the efficiency of our proposed model. Secondly, we show the generalization of our model architecture for domain adaptation on TechQA. Our experiments show that our model pre-trained on NQ can be effectively adapted to TechQA outperforming a standard fine-tuned model trained on a large PLM such as RoBERTa. To summarize, our contributions include:

1. We introduce a novel and effective yet simple paragraph representation.
2. We introduce soft labels to leverage information from local contexts near ground-truth during training which is novel for MRC.
3. Our model provides similar performance to a SOTA system on NQ while being 16 times faster and also effectively adapts to a new domain: TechQA.

2 Related Work

Machine reading comprehension has been widely modeled as cloze-type span extraction (Chen et al., 2017; Cui et al., 2017; Devlin et al., 2019). In NQ, we need to identify answers in two levels, long and short answers. (Alberti et al., 2019a) adapt a span extraction model for short answer extraction. (Zheng et al., 2020; Liu et al., 2020) construct complex networks for paragraph-level representation to enhance long answer classification along with span extraction for short answers. In this work, we propose a more light-weight and parallel-efficient way for constructing paragraph-level representation and classification by using longer context and

modeling the negative instance through Gaussian prior optimization.

Using the hierarchical nature of a long document for question answering has been previously studied by (Choi et al., 2017), where they use a hierarchical approach to select candidate sentences and extract answers in those candidates. However, due to the limit of input length for large PLMs, existing methods (Alberti et al., 2019b; Zheng et al., 2020; Chakravarti et al., 2020) slice long documents into document pieces and perform prediction for each piece separately. In our work, we show that by modeling longer input with position-aware paragraph representation coupled with Gaussian prior optimization (which is novel for MRC), we can achieve comparable performance using much simpler architecture compared to previous models, which coincide with recent new PLM for long inputs on question answering (Ainslie et al., 2020)[1].

3 Model Architecture

In this section, we introduce VAULT, our proposed model that uses a simple yet effective paragraph representation based on a longer context. VAULT starts from a base classifier that utilizes position-aware paragraph representations trained on top of a large PLM: Longformer (Beltagy et al., 2020). Next, we further introduce our Gaussian Prior-based training objective that considers partial credits for positions near the ground-truth, instead of only focusing on one ground-truth position. We show an overview of VAULT on the example from Figure 1 in Figure 2.

3.1 A Base "Paragraph" Predictor Model

SOTA methods for paragraph prediction (Zheng et al., 2020; Liu et al., 2020) represent paragraphs through expensive graph modeling, making it inefficient for "large-scale" production MRC systems. On the other hand, simply selecting the first paragraph performs poorly (Kwiatkowski et al., 2019). We hypothesize that by modeling a much longer context even simple paragraph representation can be effective for paragraph (i.e., long answers) classification. For this purpose, we employ a large-window PLM: Longformer (Beltagy et al., 2020), which has shown effectiveness in modeling long contexts for QA (Yang et al., 2018; Welbl et al., 2018; Joshi et al., 2017). Compared to conventional Transformer-based PLMs e.g. RoBERTa (Liu et al., 2019) that can only take up to 512 subword tokens, Longformer provides a much larger maximum input length of 4,096.

Position-aware Paragraph Representation (PAPR): To address the fact that many popular *unstructured* texts such as Wikipedia pages have relatively standard ways of displaying certain relevant information (e.g. birthdays are usually in the first paragraph vs. spouse names are in the "Personal Life" paragraph), we provide the base model with a representation of which part of the text it is reading by marking the paragraphs with special atomic markup tokens (`[paragraph=i]`) at the beginning of each paragraph, indicating the position of the paragraph within the text[2]. With this input representation, we then directly perform long answer classification using the special paragraph token output embedding. Formally, for every paragraph $l_i \in P$, where P are all paragraphs in a text and the representation for the corresponding markup token $\boldsymbol{h}_i^p$, the logit of a paragraph answer a it computes is as $a_i^p = \boldsymbol{W}\boldsymbol{h}_i^p + \boldsymbol{b}$.

We obtain additional document-piece representation from the standard `[CLS]` (Devlin et al., 2019) token to model document pieces that do not contain paragraph answers. The probability of choosing the paragraph given context $\boldsymbol{c}$, is computed as the softmax over paragraph candidate (with an answer span) logits and not containing answer logit:

$$p_l(l_i \mid \boldsymbol{c}) = \text{softmax}(a_i^p).$$

We pad the paragraph representations to ensure a rectangular tensor in a batch. Our final prediction strategy is similar to Zheng et al. (2020) as we first choose the paragraph candidate with the highest logit among all candidates. We then extract span answers within the selected paragraph answer candidate using a standard pointer network.

3.2 Gaussian Prior Optimization (GPO)

Conventional span extraction models (Chen et al., 2017; Glass et al., 2020; Liu et al., 2020) optimize the probability of predicted start and end positions of the answer spans with ground-truth spans via maximum likelihood estimation (MLE,Wilks et al., 1938). MLE methods promote the probability for the ground-truth positions while suppressing the probability for all other positions. However we hypothesize that, for all those negative instances, the positions that are near the ground-truth should

[1]The code and model weights of ETC has not been released at the time of writing of the paper for us to have an accurate comparison.

[2]Similar tags are added for lists and tables.

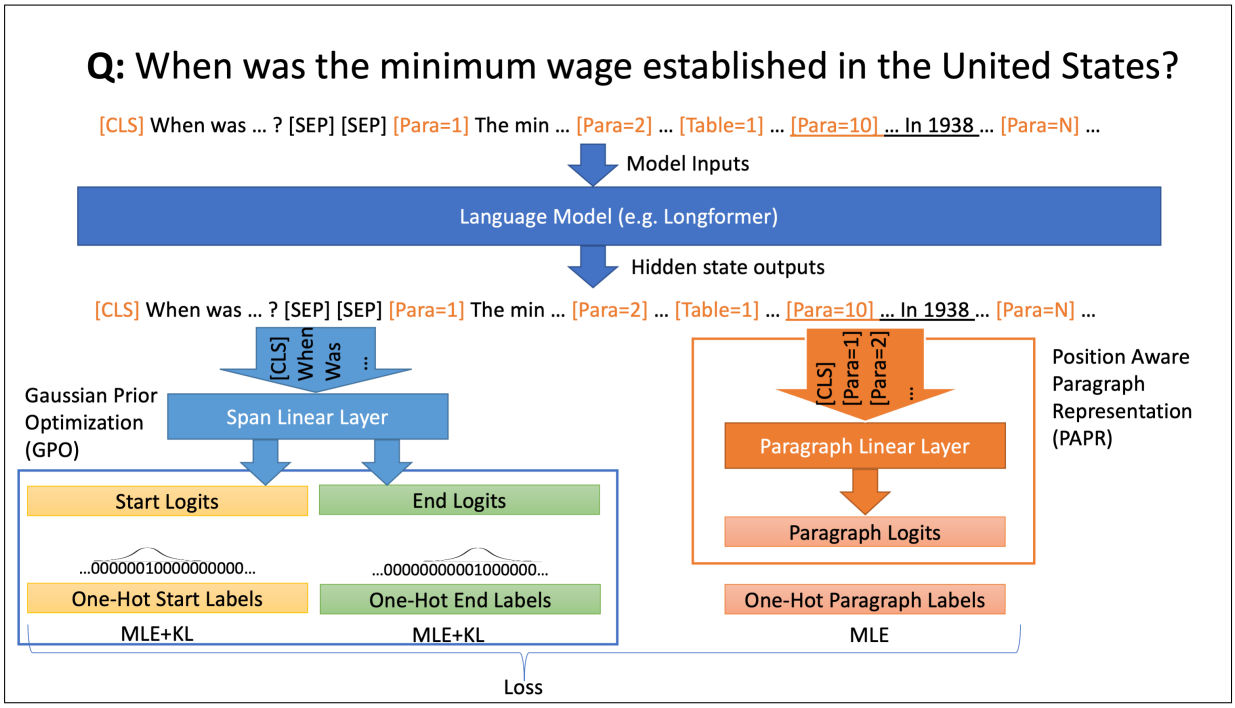


Figure 2: Overview of VAULT answering the example from Figure 1. The 10th paragraph containing the correct answer is underlined. The span linear layer receives hidden state outputs from all 4096 tokens in the window to create the start and end logits. The paragraph linear layer receives the orange-highlighted [CLS] and markup tokens (e.g. [Para=10]) to predict in which paragraph the answer resides. These logits are then used together to first select the best paragraph (LA) and finally select the best answer within said paragraph (SA).

be given higher credit than farther distant positions, since the extracted answers will be partially overlapping with the ground-truth.

To tackle this problem, we follow the intuition from Li et al. (2020) which proposes to promote the probability of generating synonyms using a Gaussian-like distribution for machine translation. We construct the distribution where it has the highest probability at ground-truth positions, and drop the probability exponentially as computed by the distance to the corresponding ground-truth positions. Specifically, for a groundtruth start or end position at y_s, where $s \in \{\text{start}, \text{end}\}$, we use a Gaussian distribution $\mathcal{N}(y_s, \sigma)$, where the mean is the position y_s and variance σ is a hyperparameter. We consider the probability density $\varphi(y \mid y_s, \sigma)$ of the Gaussian distribution at each position y as the logit for the corresponding position. We then use the softmax function with temperature T to rescale the logits to get the Gaussian-like distribution $q(y \mid \hat{y}_s)$ for ground-truth distribution at position y_s,

$$q(y \mid y_s) = \text{softmax}(\varphi(y \mid y_s, \sigma)/T).$$

We augment our MLE objective with an additional KL divergence (Kullback and Leibler, 1951) term between constructed distribution $q(y \mid y_s)$ and model prediction $p_s(y \mid \boldsymbol{c}), s \in \{\text{start}, \text{end}\}$, so that we can guide our model to follow the Gaussian-like distribution for partial credit.

$$\begin{aligned} L_D &= KL\left(q(y \mid y_s) \parallel p_s(y \mid \boldsymbol{c})\right) \\ &= \sum_y q(y \mid y_s) \log p_s(y \mid \boldsymbol{c}) \\ &\quad - \sum_y q(y \mid y_s) \log q(y \mid y_s). \end{aligned}$$

We refer to this final model as VAULT.

4 Experiments

Datasets: We experiment with two challenging "natural" MRC datasets: NQ (Kwiatkowski et al., 2019) and TechQA (Castelli et al., 2020). We provide a brief summary of the datasets and direct interested readers to the corresponding papers. NQ consists of crowdsourced-annotated *full* Wikipedia pages which appear in Google search logs with two tasks: the start and end offsets for the short answer (SA) and long answer (LA, eg. paragraph) – if they exist. TechQA is developed from real user questions in the customer support domain where each question is accompanied by 50 documents –

at most one of which has an answer – with answers significantly longer (~3-5x) than standard MRC datasets like SQuAD. We report official F1 scores for each dataset.

Results on NQ: We train VAULT on NQ – predicting the paragraph and span answers as NQ's LA and SA respectively – and compare against ROBERTA$_{DM}$: a RoBERTa (Liu et al., 2019) variant of the SOTA document model (DM) (Zheng et al., 2020) using the base variants for a more systematic comparison. Although it may seem fair to include a Longformer DM baseline in our table, doing so would be infeasible (and unwise) due to production resource constraints. We further show the impact of VAULT by providing ablation experiments where its components (GPO and PAPR) are removed. The base LM (Longformer in our experiments) without GPO and PAPR, is implemented in the style of (Alberti et al., 2019b; Chakravarti et al., 2020) where we first predict the SA and then select the enclosing LA. We aim to show that our proposed method provides comparable results to ROBERTA$_{DM}$ while being considerably faster while decoding and displaying improved performance over experiments just using the language model. To do this we consider development set SA and LA F1 (the F1 metrics with respect to the span and paragraph answers respectively) as well as decoding time t_{decode} (on a V100) as metrics.

Table 1 shows the results on the NQ dev set. We see VAULT and ROBERTA$_{DM}$ provide comparable F1 performance (precision and recall are shown in the Appendix). However, when it comes to decoding time, we can find VAULT decodes over 16 times faster than ROBERTA$_{DM}$. We additionally see in the ablation experiments that our enhancements increase both F1 metrics by multiple points, at the expense of some decoding time. In particular we note that the F1 performance of Longformer is not competitive with VAULT. We conclude that VAULT provides the best balance of F1 and decoding time as it is effectively tied on F1 (with ROBERTA$_{DM}$) and is only around 20 minutes slower to decode than the quickest model.

Domain Adaptation: Results on TechQA: Since VAULT has shown to be effective on NQ, we evaluate it on a new domain, TechQA. We compare it against a RoBERTa base model trained with the same hyper-parameters as (Castelli et al., 2020) – except we use 11 epochs instead of 20. We chose base instead of large (as is used for the TechQA baseline) to give a fair comparison since we are using a base PLM for our experiments with VAULT. Similarly, we use RoBERTa rather than BERT as it is closer to Longformer. Having already established the run-time effectiveness of VAULT on NQ, we focus on F1 metrics here, including "has answer" (HA) F1. We consider HA F1 our primary metric as we are exploring paragraph answer extraction in this work and (as previously mentioned) answers in TechQA are much longer than other datasets. We believe that the improvements in HA F1, at least partially, come from GPO.

Model	SA F1	LA F1	t_{decode}
ROBERTA$_{DM}$	**52.2**	70.1	11h
VAULT	51.6	**70.4**	40m
- GPO	49.1	67.6	41m
- PAPR (Longformer)	49.5	65.6	**22m**

Table 1: Comparison of VAULT vs. ROBERTA$_{DM}$ on NQ. We achieve comparable performance while being 16 times faster.

Model	F1	HA F1
RoBERTa	48.6	7.6
VAULT	**49.3**	**16.1**

Table 2: Results on TechQA dev set. VAULT clearly outperforms RoBERTa on both F1 and Has Answer F1.

Results on TechQA are reported in Table 2. We see that our VAULT model provides an improvement of 0.7 F1 and 8.5 HA F1 (denotes Has Answer); thus showing the effectiveness of our approach. In particular, we see that this approach of imputing a paragraph structure to classify provides a large boost to performance when a non-null answer exists (HA F1).

5 Conclusions

In this work we introduce and examine a powerful yet simple model for reading comprehension on long texts which we call VAULT, based on the hypothesis that with a large sequence length long answers can be classified effectively without computationally heavy graph-based models. We validate our approach by showing it yields F1 scores competitive with heavier methods at a fraction of the decoding cost on two very different domain benchmark datasets that require reading long texts.

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. 2020.

ETC: encoding long and structured data in transformers. *CoRR*, abs/2004.08483.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019a. Synthetic QA corpora generation with roundtrip consistency. *CoRR*, abs/1906.05416.

Chris Alberti, Kenton Lee, and Michael Collins. 2019b. A BERT baseline for the natural questions. *CoRR*, abs/1901.08634.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, J. Scott McCarley, Mike McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John F. Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avirup Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The techqa dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1269–1278. Association for Computational Linguistics.

Rishav Chakravarti, Anthony Ferritto, Bhavani Iyer, Lin Pan, Radu Florian, Salim Roukos, and Avi Sil. 2020. Towards building a robust industry-scale question answering system. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 90–101, Online. International Committee on Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, Vancouver, Canada. Association for Computational Linguistics.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G P Shrivatsa Bhargav, Dinesh Garg, and Avi Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782, Online. Association for Computational Linguistics.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6751–6761, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020. Data-dependent gaussian prior objective for language generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. RikiNet: Reading Wikipedia pages for natural question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6762–6771, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

J. William Murdock, Lin Pan, Chung-Wei Hang, Mary Swift, Zhiguo Wang, Chris Nolan, Prathyusha Peddi, Nisarga Markandaiah, Eunyoung Ha, Kazi Hasan, and et al. 2018. Engineered ai still matters for question answering. *Advances in Cognitive Systems*, 6:140–158.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *EMNLP*.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. *CoRR*, abs/2101.11272.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

SS Wilks et al. 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. Document modeling with graph attention networks for multi-grained machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6708–6718, Online. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434.

A Additional Experimental Results

For interested readers we further show precision and recall numbers for the NQ experiments in Table 3.

B Implementation Details

B.1 NQ

All models for this work are implemented in (Wolf et al., 2020). We use the following hyperparameters for VAULT when finetuning on NQ: sequence length 4096, doc stride 2048 (Ainslie et al., 2020), negative instance subsampling rates (has answer/no answer) 0.02/0.08, learning rate 5e-5, and 4 epochs of training.

B.2 TechQA

While TechQA does provide full HTML for its Technotes, the answers are annotated with respect to the cleaned plaintext. Therefore to determine paragraph breaks for VAULT we split on the "\n\n" token "ĊĊ" in the vocabulary. By imputing paragraph answers in this way, we are then able to predict the paragraph answer and then a contained span answer.

C Example Analysis

We examine additional examples in Figure 3 to provide insight on the improvements of VAULT. We

Model	SA F1	SA P	SA R	LA F1	LA P	LA R
RoBERTa$_{DM}$	**52.2**	57.2	**48.0**	70.1	69.4	70.9
Vault	51.6	**61.5**	44.4	**70.4**	**69.5**	**71.4**
- GPO	49.1	57.6	42.7	67.6	67.0	68.1
- PAPR (Longformer)	49.5	56.4	44.2	65.6	62.4	69.3

Table 3: Comparison of Vault vs. RoBERTa$_{DM}$ on NQ with precision (P) and recall (R) statistics.

compare the correct answers produced by Vault with the incorrect answers produced by the ablated model from the last row of Table 3 (NQ) and Roberta baseline from the first row of Table 2 (TechQA).

In the first example the gold SA is null, however there is a gold LA. This indicates that there is no short span which answers the question: the correct answer here is an entire paragraph. This does not confuse Vault which is able to identify the correct answer directly. However the ablated model which attempts to predict SA first struggles here – predicting the incorrect LA – as there is no gold SA.

In the second example we see that in this Technote both the correct and incorrect answers are single sentence paragraphs surrounded by paragraph breaks. Our Vault is able to identify the correct paragraph using our imputed structure and select the correct answer – whereas the Roberta baseline selects a nearby but incorrect answer.

Example A1 (NQ)
Question: why did government sponsored surveys and land acts encourage migration to the west
Wikipedia Page: Homestead Acts
Text: ...
An extension of the Homestead Principle in law, the Homestead Acts were an expression of the " Free Soil " policy of Northerners who wanted individual farmers to own and operate their own farms, as opposed to Southern slave-owners who wanted to buy up large tracts of land and use slave labor, thereby shutting out free white men.

The first of the acts, the Homestead Act of 1862 , opened up millions of acres. Any adult who had never taken up arms against the U.S. government could apply. Women and immigrants who had applied for citizenship were eligible. The 1866 Act explicitly included black Americans and encouraged them to participate, but rampant discrimination slowed black gains. Historian Michael Lanza argues that while the 1866 law pack was not as beneficial as it might have been, it was part of the reason that by 1900 one fourth of all Southern black farmers owned their own farms. [1]
...

Example A2 (TechQA)
Question: Are there any probes that can connecto the Nokia NSP EPC v17.9 and Nokia NSP RAN v17.3 using JMS/HTTP?
Text: release notice; downloads; nco-p-nokia-nfmp; Probe for Nokia Network Functions Manager for Packet NEWS

ABSTRACT
This new probe will be ready for downloading on July 20, 2017.

CONTENT

This probe is written to support Nokia Network Functions Manager for Packet release 17.3.

You can download the package you require from the IBM Passport Advantage website:

www-01.ibm.com...

Figure 3: Additional Examples of questions in the NQ and TechQA datasets. Vault's correct answer is shown in green, incorrect baseline in red. (A1) The correct answer is a paragraph LA; only Vault identifies the correct LA directly even though the gold SA is null. (A2) Vault identifies the correct "paragraph" answer.

Avoiding Overlap in Data Augmentation for AMR-to-Text Generation

Wenchao Du
University of California, Santa Cruz
duwc2013@gmail.com

Jeffrey Flanigan
University of California, Santa Cruz
jmflanig@ucsc.edu

Abstract

Leveraging additional unlabeled data to boost model performance is common practice in machine learning and natural language processing. For generation tasks, if there is overlap between the additional data and the target text evaluation data, then training on the additional data is training on answers of the test set. This leads to overly-inflated scores with the additional data compared to real-world testing scenarios and problems when comparing models. We study the AMR dataset and Gigaword, which is popularly used for improving AMR-to-text generators, and find significant overlap between Gigaword and a subset of the AMR dataset. We propose methods for excluding parts of Gigaword to remove this overlap, and show that our approach leads to a more realistic evaluation of the task of AMR-to-text generation. Going forward, we give simple best-practice recommendations for leveraging additional data in AMR-to-text generation.[1]

1 Introduction

Deep learning has made remarkable progress in many areas of natural language processing, including language generation (Sutskever et al., 2014; Luong et al., 2015) and semantic parsing (Dong and Lapata, 2016). Nevertheless, neural models are usually data-hungry, and sophisticated use of data augmentation can often go a long way (Konstas et al., 2017; Wang et al., 2018; Du and Black, 2019; Wei and Zou, 2019). One common method of data augmentation is to leverage large amounts of out-of-domain data for semi-supervised learning. However, without proper examination of the data being used, the external data may contain significant overlap with the test set, leading to unfair gains as a result. This issue is a unique problem for natural language generation (NLG) tasks with data augmentation, because training with data that overlaps with the test set is akin to training on the answers. In this work, we study the task of AMR-to-text generation and scrutinize the datasets used for training and evaluation. Our contributions are two-fold: (1) we develop an examination procedure to confirm that there are serious overlaps between one of the AMR datasets and Gigaword (Parker et al., 2011), and conduct experiments showing that some of the performance gains are indeed "unfair"; (2) we propose several strategies to apply when collecting external data for training, and empirically show that these strategies can mitigate the aforementioned unfair gains. For best practice, we suggest future work on AMR-to-text generation exclude Gigaword articles that are written in the nearby months of those covering Proxy to be on the safer side (strategy `no-3Months` described in Section 5).

2 Related Work

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) has gained growing interest as a semantic formalism. The first AMR-to-text generator was developed using tree transducers (Flanigan et al., 2016). More recent work heavily adopted neural models, explored different architectures, and commonly employed Gigaword data to boost results (Konstas et al., 2017; Song et al., 2018; Wang et al., 2020). The most common approach is to use JAMR (Flanigan et al., 2014) to bootstrap labels for the additional data and then add them to the training data.

Prior work on AMR generation has used automatic metrics such as BLEU (Papineni et al., 2002) and human evaluations (May and Priyadarshi, 2017). Currently, there is increased research on evaluation metrics for NLG (Zhang et al., 2019;

[1] Our code for these best practices is available at https://github.com/jlab-nlp/amr-clean.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1043–1048
August 1–6, 2021. ©2021 Association for Computational Linguistics

Dataset	# Sentences	Domain
Bolt	133	Web
Consensus	100	News
DFA	229	Web
Proxy	823	News
Xinhua	86	News

Table 1: The number of test sentences and domain of each AMR dataset. Note that LDC2015E86 and LDC2017T10 have identical test sentences.

Sellam et al., 2020, inter alia). However, we are not aware of prior work investigating the problem of test set overlap when using data-augmentation methods for generation. Closest to our work is prior practice in machine translation evaluation of excluding articles from the same time period as the test set (NIST, 2012).

3 Origin of AMR and Gigaword Overlap

In this section, we describe the reason for the overlap between the AMR dataset and Gigaword. In standard LDC releases of AMR, for example LDC2015E86 and LDC2017T10, the dev and test set consist of 5 datasets from different sources. Information about these datasets are listed in Table 1. Each sentence in the dev and test set is associated with an ID. The sentences of the Proxy dataset, in particular, have IDs that can be traced back to Gigaword articles. Upon inspection, these sentences appear to originate as close edits of sentences in Gigaword. For example, the sentence with ID "PROXY_LTW_ENG_20070831_0072.1" is originated from the Gigaword article with ID "LTW_ENG_20070831". The date on which a Gigaword news article was written is included in the ID. Since Proxy takes up more than half of the test sentences, such overlap could have a high impact on the evaluation of AMR-to-text generators. In the next section, we describe our procedure to empirically examine the effect of overlap between Proxy and Gigaword.

4 Measuring Overlap

We use the following procedure to quantitatively examine the overlap between Proxy and Gigaword dataset. For each Proxy sentence in the validation and test split, we find the Gigaword article whose ID is associated with the Proxy sentence ID. Then we tokenize and split the article into sentences. We measure the overlap between the Proxy sentence and each of the Gigaword sentences with 3 different metrics: (1) absolute count of common words, which is the number of distinct words that appear in both sentences, (2) BLEU score, and (3) ROUGE-L score.

	Mean	Median
Count 1st	13.85	13.0
Count 2nd	7.87	8.0
Count 3rd	7.16	7.0
ROUGE 1st	0.64	0.68
ROUGE 2nd	0.33	0.35
ROUGE 3rd	0.29	0.32
BLEU 1st	0.39	0.36
BLEU 2nd	0.07	0.05
BLEU 3rd	0.04	0.01

Table 2: The mean and median of the 3 highest scores for word count, BLEU, and ROUGE.

5 Exclusion Strategies

We propose and investigate 3 sampling strategies for constructing semi-supervised training datasets from Gigaword, and these strategies differ by how to exclude certain Gigaword articles: `no-ID` excludes articles whose id appeared in the proxy dataset; `no-Month` excludes articles that are written in the same month as those excluded by `no-ID` ; `no-3Months` excludes articles that are written in the same month or neighboring months from those excluded by `no-ID` . We use reservoir sampling (Vitter, 1985) to sample sentences from Gigaword. We first collect a set with 200k sentences without any exclusion as a baseline. We then filter out sentences that are from articles excluded by `no-ID` , and sample same number of sentences as those being filtered from articles that are included by `no-ID` . This yields a set of 200k sentences representing `no-ID` . We collect the sample sets for `no-Month` and `no-3Months` based on the baseline set in a similar fashion.[2]

We use the GGNN-dual-encoder model by (Ribeiro et al., 2019) as our model to study the effects of different exclusion strategies. For each exclusion strategy, we obtain 3 different samples using different random seeds and repeat experiments. We keep most of the hyperparameters from the original paper. We adjusted the learning rate schedule to accommodate larger sets of training

[2]Our code for doing this filtering is available on our GitHub repository.

	Sentence	Score
Count 1st	At least one of those bands appears to be splitting into at least two different groups.	13
Count 2nd	Even though the Bush White House has generally entrusted government agencies to officials ...	7
Count 3rd	The rentals violated U-Haul's rule requiring the tow vehicle to be at least 750 pounds heavier than the one being towed.	7
Bleu 1st	At least one of those bands appears to be splitting into at least two different groups.	0.70
Bleu 2nd	At least one of those inspections would have come at a particularly delicate time ...	0.20
Bleu 3rd	... as well as other outside organizations, at least one of which then sold tickets to its own members.	0.19
Rouge 1st	At least one of those bands appears to be splitting into at least two different groups.	0.91
Rouge 2nd	For at least a few of those percentage points, we have to thank Sheehan.	0.44
Rouge 3rd	At least one Democratic member of the group questioned Giuliani's decision to quit.	0.4

Table 3: Examples of top matches found in Gigaword with test set sentence "At least one of those bands appears to be splitting into different groups."

	No Extra Data	Top 1 (Cheat)	Top 2 to 4	Top 5 to 7	Top 8 to 10
Overall	27.58	32.71	31.67	30.82	30.85
Bolt	17.36	18.59	18.54	18.80	20.36
Consensus	20.18	21.50	22.73	21.90	22.58
Dfa	21.45	22.86	24.39	23.05	22.87
Proxy	31.56	39.12	36.85	35.75	35.68
Xinhua	25.22	24.22	26.03	27.16	25.90

Table 4: Evaluation results (BLEU) when the model is trained on cheat set and other highly overlapping sets.

	No Extra Data	Cheat	Baseline Strategy	`no-ID`	`no-Month`	`no-3Months`
Overall	24.32	30.73	32.72	32.69	31.83	32.35
Bolt	15.11	15.31	19.85	20.21	18.98	19.79
Consensus	17.04	16.47	25.02	20.80	24.55	24.00
Dfa	18.21	17.95	20.14	21.50	20.34	19.96
Proxy	29.33	38.16	38.52	38.46	37.33	37.99
Xinhua	23.01	22.52	32.21	31.82	31.08	32.65

Table 5: Results (BLEU) on LDC2015E86. Average of 3 experiments are reported.

	No Extra Data	Cheat	Baseline Strategy	`no-ID`	`no-Month`	`no-3Months`
Overall	27.58	32.71	34.46	33.53	33.44	33.16
Bolt	17.36	18.59	21.37	21.20	22.66	19.7
Consensus	20.18	21.50	25.96	27.18	26.44	25.06
Dfa	21.45	22.86	24.78	22.81	24.79	23.61
Proxy	31.56	39.12	39.81	38.84	38.09	38.39
Xinhua	25.22	24.22	32.59	31.68	31.77	32.40

Table 6: Results (BLEU) on LDC2017T10. Average of 3 experiments are reported.

data. With sample sets of 200k sentences, each experiment takes 3 days to finish on a Tesla V100.

6 Results

6.1 Overlap between Proxy and Gigaword

In this section, we measure the overlap between Proxy and Gigaword using word and n-gram overlap evaluation measures, and study the effect of the overlap on the final trained system. We list the mean and median of the 3 sentences with highest overlap scores for each overlap measure in Table 2. It is clear that sentences with the top overlap score overlap significantly more than those sentences at the 2nd and 3rd place. Examples for illustration are given in Table 3. All three metrics tend to find the same top matching sentence. Most of the time, the test sentence in Proxy is an extractive summarization or rephrase of the top match in Gigaword, indicating a concerning overlap between Proxy and Gigaword.

To investigate the impact of semi-supervised training with these Gigaword sentences that are close duplicates of the test set, we create various sets for semi-supervised training. We create a cheat set using sentences with highest matching ROUGE scores, called Top 1 (Cheat). We are also interested in the impact of sentences from the same article as these duplicates, but with less overlap. We create additional sets with those that have top 2-4 overlap scores, top 5-7 overlap scores, etc. We trained the model with these sample sets for semi-supervised training, and the results on LDC2017T10 are listed in Table 4. The cheat set helped the evaluation on Proxy by more than 7 points, but only helped other datasets by about 1 point, if not hurting. As the matching scores decrease, the improvement on Proxy also went down. This indicates that the overlap sentences between Proxy and Gigaword give a significant unfair advantage, especially for the sentences with highest overlap.

6.2 Exclusion Strategies for Gigaword

To find a good exclusion strategy for constructing semi-supervised datasets from Gigaword, we sample semi-supervised training sets as described in Section 5 and ran experiments. The results on LDC2015E86 and LDC2017T10 are presented in Table 5 and 6, respectively. The results on LDC2017T10 is generally better than LDC2015E86, since the size of training of the former is larger than that of the later. Without excluding (i.e. baseline strategy), the results on Proxy are significantly better than no additional semi-supervised data (by about 8 points on LDC2017T10 and 10 points on LDC2015E86). It is also slightly better than being trained with the cheat set. This is because training on sample sets of size 200k yields much better language model than the small cheat set. On the other hand, training on the cheat set is almost as good as training on 200k additional data, since neural models are good at memorization. For LDC2017T10, filtering out articles covering Proxy test sentences decreases performance on Proxy by 1 point; excluding articles written in the same month and nearby months further decreases results on Proxy by more than 0.5 points. For LDC2015E86, excluding articles written in the same month decreases results on proxy by more than 1 point.

	Proxy	All Other
LDC2015/no-ID	0.400	0.379
LDC2015/no-Month	**0.045**	0.200
LDC2015/no-3Months	0.387	0.192
LDC2017/no-ID	0.202	0.357
LDC2017/no-Month	**0.002**	0.129
LDC2017/no-3Months	**0.047**	0.100

Table 7: P-values from statistical tests comparing system performance against baseline sampling. Significant results at $p = .05$ are highlighted.

Finally, we perform statistical tests with a paired t-test for comparing performance of systems trained on different sample sets against the baseline (no filtering). See Table 7. For LDC2015E86, `no-Month` resulted in lower BLEU scores on proxy dataset that are statistically significant; for LDC2017T10, both `no-Month` and `no-3Months` resulted in lower BLEU scores on proxy and the differences are statistically significant. All strategies performed similarly on all other datasets. This shows that the exclusion of certain overlapping articles in Gigaword has significant impact on the evaluation on Proxy dataset, but less so on the rest.

7 Conclusion and Recommendation

In this paper, we examined Gigaword, the commonly used dataset for improving AMR-to-text generation, and found sentences that almost duplicate the test set of Proxy, one of the AMR datasets. We developed a procedure that utilizes a word overlap measure to find overlapping sentences, and

found several metrics that may be good at finding duplicating sentences. We proposed 3 different strategies for excluding overlapping data from Gigaword, and validated the idea that without filtering certain articles, the evaluation results may be unfair. For best practice, we suggest future work on AMR-to-text generation exclude Gigaword articles that are written in the nearby months of those covering Proxy to be on the safer side (`no-3Months`). Additionally, we suggest future work report results on each AMR dataset separately so that techniques favoring one dataset can be detected.

Acknowledgments

This research was supported in part by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805. The opinions expressed are those of the authors and do not represent views of the NSF.

References

NIST Open Machine Translation 2012 Evaluation Plan (OpenMT12). https://www.nist.gov/system/files/documents/itl/iad/mig/OpenMT12_EvalPlan.pdf. Accessed: 2020-02-01.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43.

Wenchao Du and Alan W Black. 2019. Boosting dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 38–43.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime G Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 731–739.

Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Jonathan May and Jay Priyadarshi. 2017. Semeval-2017 task 9: Abstract meaning representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 536–545.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. *Google Scholar*.

Leonardo FR Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing amr-to-text generation with dual graph representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3174–3185.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27:3104–3112.

Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. Amr-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Weakly-Supervised Methods for Suicide Risk Assessment: Role of Related Domains

Chenghao Yang[1], **Yudong Zhang**[1], **Smaranda Muresan**[1,2]
Department of Computer Science, Columbia University[1]
Data Science Institute, Columbia University[2]
yangalan1996@gmail.com
{zhang.yudong, smara}@columbia.edu

Abstract

Social media has become a valuable resource for the study of suicidal ideation and the assessment of suicide risk. Among social media platforms, Reddit has emerged as the most promising one due to its anonymity and its focus on topic-based communities (subreddits) that can be indicative of someone's state of mind or interest regarding mental health disorders such as r/SuicideWatch, r/Anxiety, r/depression. A challenge for previous work on suicide risk assessment has been the small amount of labeled data. We propose an empirical investigation into several classes of weakly-supervised approaches, and show that using pseudo-labeling based on related issues around mental health (e.g., anxiety, depression) helps improve model performance for suicide risk assessment.

1 Introduction

Suicide has been identified as one of the leading causes of deaths and approximately 1.5% of people die by suicide every year (WHO et al., 2016; Fazel and Runeson, 2020). Despite years o clinical research on suicide, researoners have concluded that suicide cannot be predicted using the standard clinical practice of asking patients about their suicidal thoughts (McHugh et al., 2019). Recently, Coppersmith et al. (2018) and Nock et al. (2019) discuss the opportunities of using social media combined with natural language processing (NLP) techniques to complement traditional clinical records and help in suicide risk analysis and early suicide intervention.

To facilitate further research on automatic suicide risk assessment, Zirikly et al. (2019) proposed a shared task, where they collected user data from r/SuicideWatch subreddit and annotated it with user-level suicide risk: no-risk, low-risk, medium-risk and high-risk. By comparing the results of the participating teams in this shared task, Zirikly et al. (2019) conclude that one of the major challenges comes from the insufficient data for intermediate suicide risk levels (i.e., low risk and medium risk) rather than extreme risk levels (i.e., no risk and high risk). Matero et al. (2019) find that using a dual BERT-LSTM-Attention model to separately extract information from both SuicideWatch and Non-SuicideWatch posts together with feature engineering that includes emotion features, personality scores, user's anxiety and depression scores are important for model performance.

In this paper, instead of feature engineering or complex model architectures, we explore whether weakly supervised methods and data augmentation techniques based on clinical psychology research can help improve model performance. We explore several weakly-supervised methods, and show that a simple approach based on insights from clinical psychology research (O'Connor and Nock, 2014) obtains the best performance. This model uses pseudo-labeling (PL) on data from the subreddits r/Anxiety and r/depression, which are considered important risk factors for suicide. We also present a potential application of our model for studying the suicide risk among people who use drugs, opening the door for using NLP methods to deepen our understanding between opioid use disorder (OUD) and mental health. The code for this paper can be found at https://github.com/yangalan123/WM-SRA.

2 Methods

We focus on Task A from the CLPsych 2019 shared task "Predicting the Degree of Suicide Risk in Reddit Posts" (Zirikly et al., 2019). The goal of the task is to predict the user-level suicide risk category based on their posts in the r/SuicideWatch subreddit. Specifically, a user u_i is associated with a col-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1049–1057
August 1–6, 2021. ©2021 Association for Computational Linguistics

lection of $n(i)$ posts $C_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n(i)}\}$, where each post $x_{i,k}(1 \leq k \leq n)$ has $m(i,k)$ sentences $x_{i,k} = [s_{ik,1}, s_{ik,2}, \ldots, s_{ik,m(i,k)}]$. We need to predict $y_i \in \{a, b, c, d\}$ using C_i, where a, b, c, d represent no-risk, low-risk, medium-risk and high-risk, respectively. In the original dataset, there are 496 users in the training set and 125 users in the test sets. We further split 100 users from the training set to create the validation set. The sizes for the train/valid/test sets are 746, 173, and 186 respectively.

Data Pre-processing Following the advice in (Zirikly et al., 2019), we replace all human names and URLs in the Reddit posts with special tokens "_PERSON_" and "_URL_", respectively. We also remove punctuation and stop words besides lowercasing. Due to the limitation of GPU memory, we split those large posts to be passages with no more than 128 words[1] and make sure that the split point is not in the middle of the sentence[2]. Such passages are treated as separate posts.

Model Architecture Our architecture is a BERT (Devlin et al., 2019) model. We also experimented with other state-of-the-art pre-trained language models (PLMs), including RoBERTa (Liu et al., 2019) and XLNET (Yang et al., 2019), but found BERT to work the best and thus consider it as our baseline architecture (more details can be found in Appendix A). Each post $x_{i,k}$ is fed into BERT (Devlin et al., 2019) and we get post embedding $\vec{e}_{i,k} = \text{BERT}(x_{i,k})$. Then we do simple mean-pooling to obtain the user embedding $\vec{u}_i = \frac{\sum_{k=1}^{n(i)} \vec{e}_{i,k}}{n(i)}$. Finally, we feed $\vec{u}_i$ to a fully-connected layer and use the Softmax layer to predict the risk level probability $\tilde{P}(y_i|C_i)$. The label with the largest probability is picked as the final prediction $\hat{y}_i$. For training, the cross entropy loss $\mathcal{L}_{\text{clf}}$ is applied to optimize our model.

2.1 Weakly-supervised Methods

Task-Adaptive Pre-training Recent works (Lee et al., 2020; Gururangan et al., 2020) point out that task-adaptive pre-training (TAP) can help pre-trained language models better adapt to the target domains and can bring improvement, especially in data-poor scenarios. Specifically, we continue pre-training (e.g., masked language modeling for BERT) on a task-relevant *unlabeled corpus* and then do normal fine-tuning on the task. Our unlabeled corpus consists of all r/SuicideWatch posts (aggregated per user) from the training sets of all the tasks (A, B, C) in the shared task (Zirikly et al., 2019). There are 621 users and 138,057 posts in this unlabeled corpus. We do continued pre-training for 2 to 3 epochs and do early stopping.

Multi-view Learning Multi-view learning (Xu et al., 2013) (MVL) is one of the widely recognized semi-supervised methods. Clark et al. (2018) provides a successful example of utilizing MVL in sequential labeling tasks. The idea is to create perturbations by masking words in certain positions and requiring the model to learn the similar distribution over the complete labeled examples and the corresponding masked examples besides normal classification training. However, since ours is a user-level classification task, we cannot directly borrow the same strategy from (Clark et al., 2018) as it mainly works on sequence labeling. We propose to create perturbations $\tilde{C}_i$ based on four strategies.[3] First, for each sentence, we will randomly mask 10% of tokens (**Word-Mask**). Second, considering that users may have posts of many words, we also propose a sentence-level masking strategy (**Sent-Mask**). For each post of a single user in the training set, we would randomly mask 10% of tokens. Third, we only keep the beginning and ending sentences in each passage (**BegEd**). Usually these sentences convey the main purpose of the posts and should preserve important semantics. Forth, we use Bert-extractive-summarizer (Miller, 2019) to extract the summary for each passage (**K-Sum**). It works mainly by first encoding each sentence $s_{ik,j}$ using a PLM to a continuous-valued representation $\vec{s}_{ik,j}$ and then training a K-means clustering over $\vec{s}_{ik,j}$. Finally it will pick K sentences for each passage that are closest to the center. Empirically, we set $K = 5$.

In training, we use KL-divergence to enforce the constraint that the predicted probability on perturbed examples $\tilde{P}(y_i|\tilde{C}_i)$ should be close to the one on complete examples (i.e., $\tilde{P}(y_i|C_i)$). The

[1] The 128 maximum passage length is tuned based on the validation set for both GPU memory and better computational efficiency for large posts. We do not observe a significant performance drop without a larger passage length.

[2] We use a limited-size stack and greedily add each sentence into the stack. If adding a new sentence will make the sum of lengths of all sentences in the stack exceed 128, we pop out all sentences, concatenate them to a new passage and then add this new sentence to the stack. For sentences having more than 128 words, we treat them as individual posts.

[3] The masking proportions for **Word-mask** and **Sent-Mask** are tuned empirically on the validation set.

loss incurred by KL-divergence is simply added to the classification loss and these two losses are optimized together for each training instance.

Clinical Psychology Inspired Pseudo-labeling According to the analysis of the shared task report (Zirikly et al., 2019), the main challenge for the 4-way classification comes from insufficient data for the intermediate classes (i.e., low-risk and medium-risk). A straightforward solution is to collect data for these two classes. Recent clinical psychological research (O'Connor and Nock, 2014) points out that mental health issues such as depression and anxiety can be important risk factors for suicide. Inspired by this study, we collect data from r/Anxiety and r/depression from Reddit. The time range of all collected data is from December 1, 2008 to September 30, 2020. We sample a small proportion of the collected data from both subreddits and after manual verification, we decided to assign *low-risk* labels to all r/Anxiety users in the sample and *medium-risk* labels to all r/depression users in the sample. Since we do not have experts to label these posts, adding too much pseudo-labeling data might introduce too much noise. Based on preliminary experiments on the *validation set*, the number of added pseudo-labeling data is 8% of the suicide risk assessment training data. The only difference between these experiments and the main experiments is that we only train the model for 10 epochs rather than full 20 epochs. Table 1 show results for different sizes of added pseudo-labeled data from r/depression on the validation set. All pseudo-labeling data follows roughly the same pattern with the best proportion being 8%.

$\frac{\#(\text{r/depression})}{\#(\text{Training})}$	Macro-F1 on Validation set
2%	0.408
8%	0.471
16%	0.442
32%	0.408

Table 1: Results of different proportions of added pseudo-labeling data from r/depression.

3 Experiments and Results

We implement our BERT model based on huggingface Transformer (Wolf et al., 2020). Due to the limitation of GPU memory, we only use the *base* version.We split 20% of original training data to be the validation set and fix the split for all models. The model selection is made by early stopping and we train all models for 20 epochs with the batch size 32. For users with too many posts and words, we only sample 100 passages for them. Table 2 shows our results on Macro-F1.

No.	Approach	Setup	Macro (P/R/F1)
1	Baseline	BERT	0.436 / 0.424 / 0.427
2	TAP	BERT	0.439 / 0.445 / 0.432
3	MVL	Word-Mask	0.464 / 0.466 / 0.463
4	MVL	Sent-Mask	0.380 / 0.409 / 0.383
5	MVL	BegEd	0.384 / 0.422 / 0.401
6	MVL	K-Sum	0.384 / 0.422 / 0.401
7	PL	Depression (medium-risk)	**0.535 / 0.480 / 0.498**
8	PL	Anxiety (low-risk)	0.495 / 0.469 / 0.478
9	PL	Depression + Anxiety	0.473 / 0.456 / 0.463
10	PL	Task C (low-risk)	0.475 / 0.462 / 0.460
11	-	Task C (crowd-labeled)	0.418 / 0.406 / 0.408

Table 2: Results Task A test set. For each of tasks 7-11, the size of added data is 8% of training data. Metrics are all reported on macro-average.

Task-Adaptive Pre-training After applying task-adaptive pre-training on BERT, we see small performance gains over BERT (i.e., from 0.427 to 0.432). That might be because even we use the whole corpus provided by the shared task, it is still not large enough.

Multi-view Learning Word-Mask strategy improves over the BERT baseline. Compared with the adaptive pre-training results on BERT, which also do word-level masking but only trained on language modeling, we can see that MVL provides a more efficient way to utilize a small training corpus and bring 3.1% gain on Macro-F1. However, all the other MVL approaches hurt the performance when compared to the BERT baseline. This might be because the proposed sentence-level perturbation strategy can seriously break the semantics of each post and thus influence the overall performance, and random sampling over sentences hurts most.

Clinical Psychology Inspired Pseudo-labeling Exp 7, 8 and 9 in Table 2 achieve the Top-3 Macro-F1 scores. This indicates that although our psychology-inspired pseudo-labeling technique is simpler than other weakly-supervised methods, adding meaningful pseudo-label data from relevant domains helps mitigate the problem of insufficient data in the intermediate classes (b and c). To verify this point, we show the class-wise classification results for PL-based models in Table 3 where we can

Setup	a	b	c	d
Baseline	0.730	0.077	0.333	0.566
Depression (medium-risk)	0.764	0.273	0.327	0.627
Anxiety (low-risk)	0.724	0.160	0.415	0.614
Depression + Anxiety	0.767	0.143	0.370	0.574
Task C (low-risk)	0.762	0.080	0.318	0.678
Task C (crowd-labeled)	0.667	0	0.357	0.609

Table 3: Class-wise performance (F1) for PL-based methods (a=no-risk; b=low-risk; c=medium-risk; d=high-risk).

see improvements on b and c classes. Due to space constraints, we present the class-wise performance for all models in Appendix C.

The investigation over the confusion matrix of the best model (shown in Section 4) further supports our hypothesis. However, when we try to combine different pseudo-labeling data together (see Exp 9, where we add users from r/depression and r/Anxiety following the proportion of $1:2$[4] and still keep the added user number the same), we observe a slight performance drop. The reason might be that users in these two PL datasets might be at the boundary of the low-risk and medium-risk and simply mixing them together will make the model confuse between these two classes (see Supplemental material D for all confusion matrices).

Furthermore, we wanted to test the role of the *clinical psychology* aspect of our pseudo-labeling approach. Does the gain come from the meaningful domains (anxiety and depression) or just by adding additional data? To answer this, we use additional data provided by Task C of the shared task that contains posts from random subreddits (e.g., sports). We do two experiments: 1) assign low-risk to all such users and 2) assign the gold labels provided by the task via crowdsourcing. We add the same size as for the other pseudo-label experiment (8% of training data). The results (Exp 10 & 11 in Table 2) show that the clinical psychology inspired PL outperforms these models by meaningfully addressing the intermediate classes insufficient data problem.

4 Error Analysis

In this section, we take a closer look at the prediction results of our best model (clinical psychology inspired pseudo labeling using r/depression as medium risk) by looking at the confusion matrix and sampled error cases. We plot the confusion matrices for the baseline model (Exp 1 in Table 2) and the best model (Exp 7 in Table 2) in Figure 1. We can see that, the best model achieves the improvement mainly by fixing error cases wrongly predicted as no-risk (where the true labels are "b", "c" and "d", with greater error reduction for "d") and low-risk (where the true labels are "c" and "d"). As O'Connor and Nock (2014) point out, depression is a serious mental issue and has become one of the most important risk factors of suicide. Adding posts from r/depression can help the model understand better what is "medium-risk" and "high-risk" and thus raise the alert for the signals of similar or related mental issues.

We can also see that the main problem of our best model, is still the confusion between "b" (low-risk) and "c" (medium-risk). In addition, the problem of wrongly predicting the examples belonging to intermediate classes to high-risk ones still exists. By manual investigation, we find that both problems require expertise in mental health to make the subtle distinctions. For example, the following text comes from a low-risk example[5] that is wrongly predicted as high-risk by our best model:

> " *sadness has taken me. . . i am sad , lonely , and i have no interest in living anymore. . . i didnt want to die. . . my mind is diseased , unable to take happiness. . . i have no interest in forming any more.* . . . i dont think ill do it. . . "

It can be seen that there are many negative or even desperate expressions (marked as red) in this examples, mixed with some short signals (marked as blue) possibly indicating a person considered at low-risk. The model can be fooled by the massive negative expressions and make the wrong predictions if the model is not aware of the true intent of the person. Therefore, reliable intent identification that could consider user posts across time and other information would be a powerful tool to help the model prevent mistakes like this.

5 Application: Predicting Suicide Risk of People Who Use Drugs

In order to further verify the effectiveness of our model in real-world applications, we create a sim-

[4] See Supplemental material B for detailed experiments over different mixing proportions

[5] Based on ethical consideration, we drop out many sensitive and private content of this example.

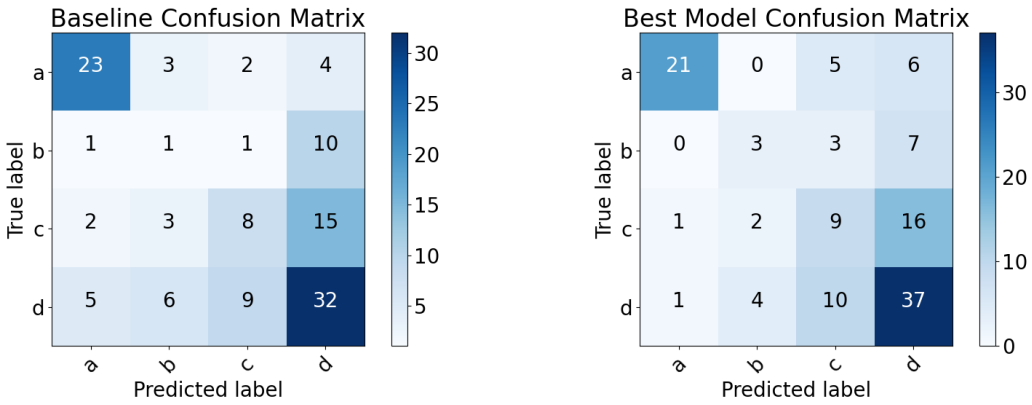


Figure 1: Visualization of the confusion matrices for the baseline model (Exp 1) and the best model (Exp 7) .

ulation scenario: we apply our best model (Exp 7) over the data that is collected for 612 users who post on both r/opiates and r/SuicideWatch. r/opiates is a subreddit where people discuss topics around opioid usage (e.g., drug doses, withdrawal anguish, daily experiences, harm reduction). This community members could often be at a high suicide risk (Aladağ et al., 2018; Yao et al., 2020). We apply our model over their $1,176$ posts on r/SuicideWatch and find that our model predicts that 15.52% of them are no-risk, while 84.48% of them are of low-risk, medium-risk and high-risk. The results on sampled $2,863$ r/opiate posts are 30.56% for no-risk and 69.44% for at least some risk. The predicted outputs are highly aligned with reported results using crowdsourcing annotation of suicidal or not-suicidal by Yao et al. (2020) and show the effectiveness of our model in this simulated scenario.[6] We hope this will open the door of using NLP methods to investigate the link between suicidal ideation and fatal overdoses among people who use drugs.

6 Conclusions

We investigated a series of weakly-supervised methods and find that pseudo-labeling on data related to risk factors for suicide (depression, anxiety) can help improve model performance. This provides an alternative way to use theoretically-grounded models (e.g., compared to feature engineering). We also show a potential use case of this work for understanding suicidal ideation among users who use drugs (e.g., opiates).

[6]The original Mturk annotation dataset is not open-sourced and thus we can only do rough trend matching on our own collected data.

Ethical Considerations

The dataset for suicide risk assessment was obtained from the organizers of the 2019 Clinical Psychology Shared Task on Suicide Risk Assessment, by filling in a participant application where we affirmed that we would follow the shared task's rules. We have obtained IRB approval (exempt) from Columbia University to use the data as it consists of publicly available and anonymous posts extracted from Reddit. For the application part, we also obtained Columbia IRB approval (exempt) for the data publicly available and anonymous data from r/opiates. All data is kept secure and online userIDs are not associated to the posts.

Our intention of developing and improving suicide risk assessment models is to help health professionals and/or social workers identify people that might be at risk of committing suicide. We emphasize our intention that suicide risk assessment models such as the ones developed here to be used responsibly, with a human in the loop — for example a medical professional, a mental health specialist, who can look at the predicted labels and offer explanations and decide whether or not they seem sensible. We would urge any user of suicide risk assessment technology to carefully control who may use the system. Currently, the presented models may fail in two ways: they may either mislabel an at-risk user as no-risk (our current models are particularly designed to minimize this risk), or classify a no-risk user with some level of risk. Obviously, there is some potential harm to a person who is truly in need if a system based on this work fails to detect their suicidal ideation, and it is possible that a person who is not truly in need may be irritated or offended if someone reaches out to them because

of a mistake. That is why, this system needs only to be used as additional help for health professionals.

We note that because most of our data were collected from Reddit, a website with a known overall demographic skew (towards young, white, American men[7]), our conclusions about what expressions of different suicide risk levels look like and how to detect them cannot necessarily be applied to broader groups of people. This might be particularly acute for vulnerable populations such as people with opioid use disorder (OUD). We hope that this research stimulates more work by the research community to consider and model ways in which different groups express suicidal ideation.

References

Ahmet Emre Aladağ, Serra Muderrisoglu, Naz Berfu Akbas, Oguzhan Zahmacioglu, and Haluk O Bingol. 2018. Detecting suicidal ideation on forums: proof-of-concept study. *Journal of medical Internet research*, 20(6):e215.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925.

Glen Coppersmith, Ryan Leary, Patrick Crutchley, and Alex Fine. 2018. Natural language processing of social media as screening for suicide risk. *Biomedical informatics insights*, 10:1178222618792860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Seena Fazel and Bo Runeson. 2020. Suicide. reply. *New England journal of medicine*, 382(21):e66–e66.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

[7] https://social.techjunkie.com/demographics-reddit

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Matthew Matero, Akash Idnani, Youngseo Son, Salvatore Giorgi, Huy Vu, Mohammad Zamani, Parth Limbachiya, Sharath Chandra Guntuku, and H Andrew Schwartz. 2019. Suicide risk assessment with multi-level dual-context language and bert. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 39–44.

Catherine M McHugh, Amy Corderoy, Christopher James Ryan, Ian B Hickie, and Matthew Michael Large. 2019. Association between suicidal ideation and suicide: meta-analyses of odds ratios, sensitivity, specificity and positive predictive value. *BJPsych open*, 5(2).

Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

Matthew K Nock, Franchesca Ramirez, and Osiris Rankin. 2019. Advancing our understanding of the who, when, and why of suicide risk. *JAMA psychiatry*, 76(1):11–12.

Rory C O'Connor and Matthew K Nock. 2014. The psychology of suicidal behaviour. *The Lancet Psychiatry*, 1(1):73–85.

WHO et al. 2016. Suicide across the world.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Hannah Yao, Sina Rashidian, Xinyu Dong, Hongyi Duanmu, Richard N Rosenthal, and Fusheng Wang. 2020. Detection of suicidality among opioid users on reddit: Machine learning–based approach. *Journal of medical internet research*, 22(11):e15293.

Ayah Zirikly, Philip Resnik, Ozlem Uzuner, and Kristy Hollingshead. 2019. Clpsych 2019 shared task: Predicting the degree of suicide risk in reddit posts. In *Proceedings of the sixth workshop on computational linguistics and clinical psychology*, pages 24–33.

A Comparison of Different Pre-trained Language Models

Given that there has been significant progress on the architecture designs after BERT, we have experimented with different PLMs, such as RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019). From Table 4, we can see that on the Test set, the Macro-F1 scores for BERT and RoBERTa are almost the same and XLNet performs worse than BERT. Therefore, we hypothesis that the architecture of PLMs will not influence substantially the results on this task so we chose BERT model.

PLM	TAP?	PL?	MVL?	Macro-F1
BERT	No	No	No	0.427
XLNET	No	No	No	0.422
RoBERTa	No	No	No	0.408

Table 4: Experiment results for different PLMs. Here we only show the macro-F1 for the baseline model built on different PLMs.

B Results for Different Mixing Proportions

Table 5 shows the results for different mixing proportions of pseudo-labeling data from r/Anxiety and r/depression. Due to the limitation of space, in the main paper, we only show the results achieved by the best mixing proportions.

Mixing Proportion	Macro-F1
1: 5	0.398
1: 2	0.463
1: 1	0.434
2: 1	0.441
5: 1	0.442

Table 5: Experiment results for different mixing proportions. Here the proportion represents the user ratio of #(r/depression) : #(r/Anxiety).

C Class-wise Decomposition of Experimental Results

Here we show the class-wise performance for all the models in Table 6.

D Additional Error Analysis

Additional confusion matrices for high-performance models (8, 9, 10 in Table 2) are in Figure 3.

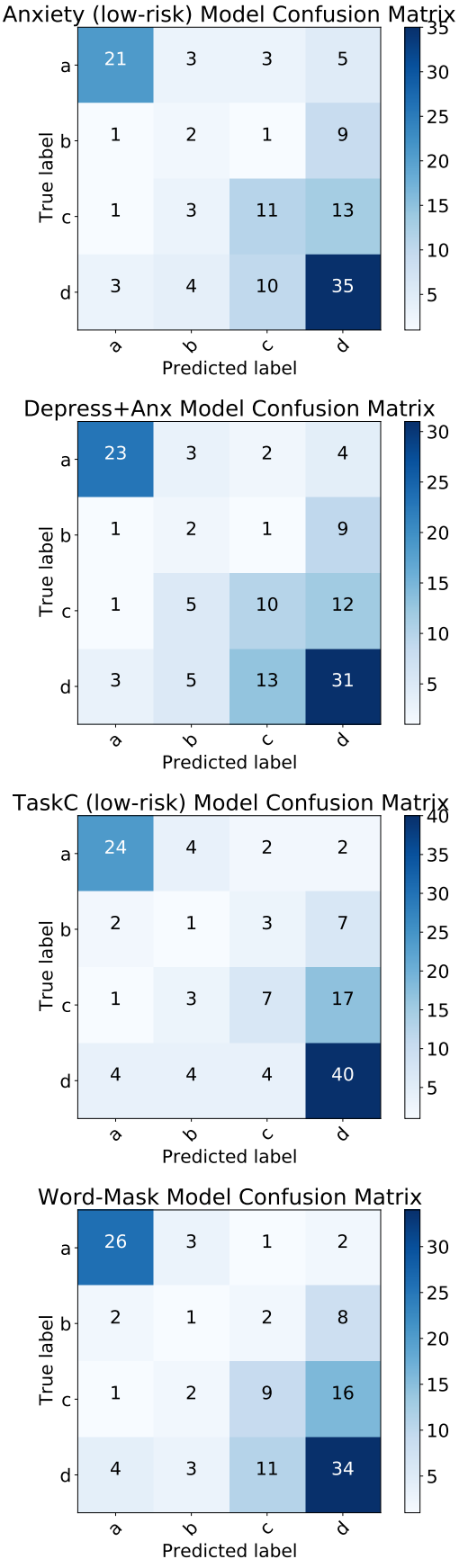


Figure 2: Word-Mask Confusion Matrix.

Figure 3: Additional Confusion Matrices for Task 8, 9, 10, 3 in Table 2

No.	Approach	Setup	a	b	c	d
1	Baseline	BERT	0.742/0.719/0.730	0.077/0.077/0.077	0.400/0.286/0.333	0.525/0.615/0.566
2	TAP	BERT	0.774/0.750/0.762	0.143/0.154/0.148	0.250/0.107/0.150	0.588/0.769/0.667
3	MVL	Word-Mask	0.788/0.812/0.800	0.111/0.077/0.091	0.391/0.321/0.353	0.567/0.654/0.607
4	MVL	Sent-Mask	0.551/0.844/0.667	0.091/0.077/0.083	0.294/0.179/0.222	0.583/0.538/0.560
5	MVL	BegEd	0.686/0.750/0.716	0/0/0	0.320/0.286/0.302	0.531/0.654/0.586
6	MVL	K-Sum	0.686/0.750/0.716	0/0/0	0.320/0.286/0.302	0.531/0.654/0.586
7	PL	Depression (c)	0.913/0.656/0.764	0.333/0.231/0.273	0.333/0.321/0.327	0.561/0.712/0.627
8	PL	Anxiety (b)	0.808/0.656/0.724	0.167/0.154/0.160	0.440/0.393/0.415	0.565/0.673/0.614
9	PL	Depression + Anxiety	0.821/0.719/0.767	0.133/0.154/0.143	0.385/0.357/0.370	0.554/0.596/0.574
10	PL	Task C (b)	0.774/0.750/0.762	0.083/0.077/0.080	0.438/0.250/0.318	0.606/0.769/0.678
11	-	Task C (crowd-labeled)	0.760/0.594/0.667	0/0/0	0.357/0.357/0.357	0.556/0.673/0.609

Table 6: Class-wise decomposition results for models considered in this paper. The results under each class are presented following the "Precision/Recall/F1" format.

Can Transformer Models Measure Coherence In Text? Re-Thinking the Shuffle Test

Philippe Laban , Luke Dai , Lucas Bandarkar , and Marti A. Hearst

UC Berkeley
{phillab, luke.dai, lucasbandarkar, hearst}@berkeley.edu

Abstract

The *Shuffle Test* is the most common task to evaluate whether NLP models can measure coherence in text. Most recent work uses direct supervision on the task; we show that by simply finetuning a RoBERTa model, we can achieve a near perfect accuracy of 97.8%, a state-of-the-art. We argue that this outstanding performance is unlikely to lead to a good model of text coherence, and suggest that the Shuffle Test should be approached in a Zero-Shot setting: models should be evaluated without being trained on the task itself. We evaluate common models in this setting, such as Generative and Bi-directional Transformers, and find that larger architectures achieve high-performance out-of-the-box. Finally, we suggest the k-Block Shuffle Test, a modification of the original by increasing the size of blocks shuffled. Even though human reader performance remains high (around 95% accuracy), model performance drops from 94% to 78% as block size increases, creating a conceptually simple challenge to benchmark NLP models.

1 Introduction

In recent years, text generation applications, fueled by Transformer models pre-trained on large datasets, have achieved dramatic results on a wide range of NLP tasks. These include GPT2 applied to story completion of fan fiction (Radford et al., 2019), the PEGASUS model (Zhang et al., 2020) improving state-of-the-art on ten summarization datasets in widely varying domains, and more recently GPT3 (Brown et al., 2020) doing well on a diversity of tasks in a zero-shot setting. However, it is not clear how *coherent* the text generated by these models is.

The computational linguistics literature holds many competing definitions of *coherence* in text; Wang and Guo (2014) provide a useful brief summary of key competing theories. This work attempts to identify the *absence* of coherence, noting that a text might be composed of valid sentences when viewed independently, but when read sequentially, semantic relations are not well-supported.

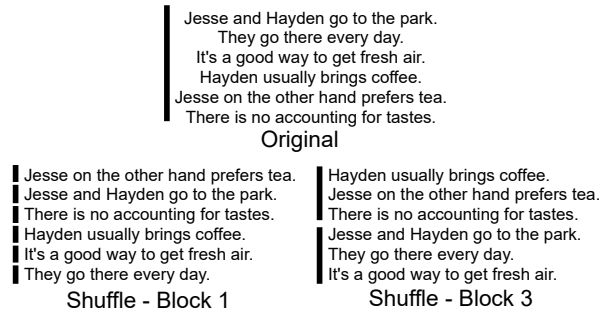


Figure 1: **Can modern NLP models recognize shuffled, incoherent text without supervision?** Yes (mostly) when all sentences are shuffled (left), but less so when shuffling k blocks at a time (right).

The NLP community has proposed models to *measure* coherence, as well as repeatable tasks to evaluate these models. In this paper, we outline these common tasks, and describe what we believe is a limitation in the framework of the most common task: the Shuffle Test. The Shuffle Test is a conceptually simple and reproducible task, in which a model must differentiate between an original text and a sentence-order shuffled version. Because of its simplicity, we make the argument that the Shuffle Test should be viewed as a *probe*: a task on which models should be evaluated without directed supervision. Prior work (Paulus et al., 2018) has shown that directly optimizing evaluation metrics such as ROUGE or BLEU leads to inadequate models, exploiting weaknesses in the evaluation metrics.

We show that this phenomenon occurs with the current application of the Shuffle Test in related work. To demonstrate the pitfalls, we finetune a RoBERTa-large model – an architecture several or-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1058–1064
August 1–6, 2021. ©2021 Association for Computational Linguistics

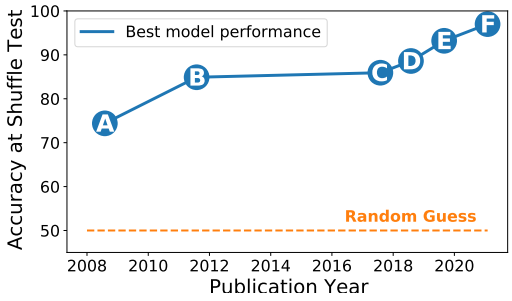


Figure 2: Timeline of incremental accuracy improvements on the *Shuffle Test* on the WSJ corpus. Letters are for models described in Section 2.2.

ders of magnitude larger than previously used models – on the Shuffle Test and show the results outperform previous models, with an accuracy of 97.8%. We argue this model has most likely learned features specific to recognizing shuffled-ness, which is probably a conflated signal for the underlying goal of learning a strong coherence model.

We first outline prior work on tasks and models to measure textual coherence, then describe the framework for the Zero-Shot Shuffle Test, showing how to adapt common models to the setting, and finally propose a variation to the Shuffle Test that significantly increases the challenge for models, while not affecting human performance at the task.[1]

2 Tasks and Models for Coherence

2.1 Tasks for Coherence Evaluation

The Shuffle Test, introduced by Barzilay and Lapata (2008), is the most common task for coherence model evaluation. The task is a binary classification, in which a model must discriminate between a document and a *shuffled document*, obtained by randomly shuffling the order of sentences in the document. The most common dataset for evaluation is a set of articles from the Wall Street Journal (Elsner and Charniak, 2011).

In **the Insertion Test**, a single sentence from a document is removed, and the model must predict the sentence position. Typically, models assign a score to each possible position, and predict the one with highest score. One limitation of the Insertion Test is that model accuracies are low, often in the 10-20% range (Elsner and Charniak, 2011). To our knowledge, there is no evaluation of human performance on this test; with the possibility that the task can have more than one plausible solution. Computational cost is another limitation, often growing linearly with the number of sentences.

In **the Sentence Ordering Task**, a model is given an randomly ordered sentence set, and must produce the correct ordering of sentences. The task is often restricted to generative models, as it is prohibitively expensive to score all combinations to extract a best-scoring order (Logeswaran et al., 2018).

2.2 Models for the Shuffle Test

Figure 2 is a timeline of models that have led to progress on the Shuffle Test since its introduction.

The Entity Grid (model A in Fig. 2) was introduced by Barzilay and Lapata (2008). A text is transformed into an entity grid, a matrix (#sentences x #entities) indicating presence of an entity in a sentence. The entity grid is featurized and used to train a predictor on coherence tasks.

Elsner and Charniak (2011) (model B) extended the entity grid by adding linguistic features such as named-entity type. Nguyen and Joty (2017) (model C) introduce the first neural approach, using a convolutional neural network (CNN) to operate over the entity grid, and Joty et al. (2018) (model D) added word embeddings to entity-grid features. Most recently, Moon et al. (2019) (model E) replaced traditional word vectors with ELMO (Peters et al., 2018) contextual word vectors.

Crucially, all these models are directly trained on the Shuffle Test, and with each iteration of improvement, model capacity (i.e., the number of trainable parameters) has increased. We finetune a RoBERTa-large (Liu et al., 2019) (model F), a still larger model, on the Shuffle Test, and achieve a 97.8% accuracy on the WSJ test set, a new state-of-the-art.

Training details. We finetune the RoBERTa-large on the training portion of the WSJ dataset, and setup the task as a sequence classification. We trained using the ADAM optimizer, using a learning rate of $1e^{-5}$ and a batch-size of 16. The model was trained on a single GPU, an Nvidia V-100, and training converged within 10 minutes. Model checkpoint was selected based on a validation set accuracy, and tested once on the standard WSJ test set.

This stellar performance leads us to believe that there are two conflated factors that cause good performance on the Shuffle Test: a model that can

[1]The code and model checkpoints are available at: https://github.com/tingofurro/shuffle_test.

truly recognize the lack of coherence in shuffled text, and a model specialized at the Shuffle Test, recognizing shuffle-specific features in text, without assessing textual coherence. This resonates with findings from Mohiuddin et al. (2020), who show that increased model performance at the Shuffle Test in supervised models does not necessarily lead to improvements in downstream tasks, such as ranking of generated summaries. Ideally, the Shuffle Test would be used to assess coherence models that work independently of the test itself.

We propose a simple solution: coherence models should be evaluated on the Shuffle Test in a Zero-Shot setting; without being supervised on the test, preventing the learning of shuffle-specific features, and more directly evaluating coherence aptitudes.

3 Zero-Shot Shuffle Test

We now define specifically what factors need to be respected to satisfy the zero-shot setting.

> In the Zero-Shot Shuffle Test, the evaluated model must not be pre-trained, fine-tuned or modified using shuffled text.

More specifically, this restricts the use of the Shuffle Test as supervision (as a binary classification task), as well as other tasks that involve shuffling text, such as the sentence ordering task.

Next, we adapt common architectures to the Zero-Shot Shuffle Test and assess performance in diverse textual domains.

3.1 Zero-Shot Coherence Models

We adapt the two common classes of Transformer models to the Zero-Shot Shuffle Test: Generative and Bi-directional Transformers.

Generative Transformers are compatible with language modeling, in which a model assigns a likelihood to a sequence of words ($S = [W_1, ...W_n]$). Transformers estimate the likelihood of a sequence by factoring on sequence order:

$$P(S) = \prod_{i=1}^{N} P(W_i|W_1...W_{i-1}) \quad (1)$$

Taking the log of the likelihood ($log(P(S))$) is often preferred as it allows for numerical stability.

To perform a Zero-Shot Shuffle Test, we compute log-likelihoods of the original and shuffled documents and predict the lower-scoring one as shuffled.

We experiment with GPT2 models of varying sizes (**GPT2-base**, **GPT2-medium**, **GPT2-large**), and finetune an In-Domain GPT2-medium using a language modeling loss in each domain to evaluate whether in-domain specialization improves performance (**GPT2-med-ID**).

When texts exceed sequence-length limits of models (e.g., 512 words), we implement a sliding window mechanism. The sequence is split into successive windows with 50% overlap. Window log-likelihoods are averaged into a document log-likelihood.

Bi-Directional Transformers, exemplified by BERT (Devlin et al., 2019), are the second class of models we adapt to the test. Unlike Generative Transformers, bi-directional Transformers do not impose strict sequence order, rendering sequence likelihood estimation less straightforward.

Salazar et al. (2020) propose a solution, with Masked Language Model Scoring (MLMS), in which a likelihood is estimated by masking each word in the sequence, predicting its identity, and averaging all word-likelihoods into a score:

$$\text{MLMS}(S) = \frac{1}{N}\sum_{i=1}^{N} \log P_{MLM}(W_i|W_{\setminus i}) \quad (2)$$

where $W_{\setminus i} = S - \{W_i\}$. Unlike generative models, each word's likelihood is conditioned on all others, an advantage of Bi-directional models. For the Zero-Shot Shuffle Test, the document with lower MLMS is predicted as shuffled.

One key disadvantage of MLMS is its computational cost: scoring requires a forward-pass for each word in the sequence; by contrast, generative models usually require a single forward pass. This limits our ability to test large models, and therefore test only base models: **BERT-base** and **RoBERTa-base**.

3.2 Datasets

To examine whether there are significant differences in performance across domains, we evaluate with the Shuffle Test using three distinct domains. We performed a manual check to determine that the datasets we selected do not overlap with the dataset used to pre-train BERT, RoBERTa and GPT2. The three domains are:

News domain. We use the standard Wall Street Journal (WSJ) test-set introduced by Elsner and Charniak (2011) in the supervised Shuffle Test. The dataset contains 1006 documents.

	Domain (%)			
Model	**WSJ**	**Legal**	**Reddit**	**Overall**
GPT2-base	47.2	92.0	74.8	71.3
GPT2-medium	91.2	98.6	88.9	92.9
GPT2-large	73.2	**99.3**	90.6	87.7
BERT-base	73.2	96.1	86.1	85.1
RoBERTa-base	82.3	94.8	**96.7**	91.3
GPT2-med-id	**93.1**	98.8	90.0	**94.0**

Table 1: Accuracy of Zero-Shot Shuffle Tests of models on three domains: Wall Street Journal (WSJ), Billsum documents (Legal), and Reddit. We report an overall, averaged performance across domains.

	Block Size				
Model	**1**	**2**	**3**	**4**	**5**
Human Perf. - WSJ	97.5	94.5	93.0	96.0	94.0
GPT2-med - WSJ	95.3	91.4	89.5	87.4	85.3
GPT2-med - Legal	98.7	98.0	96.9	95.9	94.5
GPT2-med - Reddit	89.5	76.9	66.1	59.1	53.8
GPT2-med - Avg.	94.5	88.8	84.2	80.8	77.9

Table 2: **Results of Zero-Shot KBST varying the block size from one to five.** The GPT2-medium model was tested on all three domains, and human performance was measured on WSJ.

Legal domain. We use the full document released in the BillSum dataset (Kornilova and Eidelman, 2019) which consists of US Congressional and California state bills. We use the first 1,000 documents in the standard test set.

Blog domain. We use posts of the Reddit TIFU dataset (Kim et al., 2019), consisting of stories written by members of the Reddit community. We use the first 1,000 documents of the corpus.

We choose these datasets as they are publicly available, can easily be accessed through the HuggingFace datasets package (Wolf et al., 2020) and represent a diversity of textual domains.

We note that document length affects the amount of displacement that occurs from shuffling, with more displacement in longer texts. To take this effect into account, we truncate documents at 20 sentences before administering the Shuffle Test.

3.3 Results

Overall, all models significantly outperform random chance, with the GPT2-medium achieving 91.2% on the WSJ test-set out of the box where the previous supervised state-of-the-art was 93%. Bi-directional models achieve better results than generative models at Transformer-base size (e.g., GPT2-base vs. RoBERTa-base).

Increasing model size leads to large performance improvement for GPT2, confirming that according to the Shuffle Test, larger Transformer models improve at modeling coherence. In-domain finetuning leads to an improvement on all domains (GPT2-med-id outperforming GPT2-medium), confirming the strength of in-domain finetuning (Howard and Ruder, 2018).

Finally, models achieved stronger performance on the Legal domain, with models all scoring 92.0 or above. Overall, three of the six models we test achieve compound performance over 90%.

There is a potential question about the zero-shot nature of the BERT training method. The original BERT model is trained with two objectives, one of which is Next Sentence Prediction (NSP). In NSP, the model is exposed to two blocks of text, and must predict whether they are adjacent in a document or not. It can be argued that NSP is an indirect supervision signal for the Shuffle Test. However, we find that the BERT model performs worse than RoBERTa, a similar model in architecture trained without the NSP objective. This difference in performance suggests that the NSP objective is not the cause of the superior performance of these models, thus preserving the claim that they act in a zero-shot manner for the purposes of the Shuffle Test.

We next propose a modification to the Shuffle Test that challenges models significantly more.

4 The k-Block Shuffle Test

Results in Section 3 can be interpreted to mean that with large enough Transformer models, the Shuffle Test with no supervision is essentially a solved task. We find that a simple modification of the Shuffle Test can significantly reduce model performance, without affecting human annotator performance.

The modification we propose, k-Block Shuffle Test (KBST), is illustrated in Figure 1. In the standard Shuffle Test, text is divided into sentences and shuffled, with a unit of one sentence. In the k-Block Shuffle Test, sentences are grouped in contiguous blocks of k sentences (resembling paragraphs), and the blocks are shuffled, maintaining sentence order in each block. Within a block, sentences remain locally coherent, and as block size increases, the fraction of correct sentence transitions increases, while potentially incoherent transitions decrease.

To establish the feasibility of the KBST with differing block-sizes, we performed a human evaluation completed by authors of the paper as well as a

third annotator recruited on the Upwork[2] platform. This annotator is a native English speaker with experience in proofreading, and was remunerated at $15/hour USD.

The human evaluation consisted of performing KBST on 500 documents randomly sampled from the WSJ dataset, with 100 tests for each block-size from one to five. Each Shuffle test was performed by at least two annotators. We find that there is high inter-annotator agreement (Cohen's Kappa $\kappa = 0.86$), which does not significantly vary with block-size (ranging from 0.76-0.94).

KBST results for human and computational models are shown in Table 2. Human performance is very high on WSJ, averaging above 95%, and is not significantly affected by block-size.

Timings logged during human annotation show that Shuffle Tests took on average 40% more time for larger (3-5) than smaller blocks (1-2), showing the task requires more attention from annotators as block size increases.

In all three domains, increased block size leads to a decrease in model performance. The magnitude of decrease in performance from block-size 1 to 5 is sensitive to the domain, with a drop of 4% in the legal domain, and 36% for Reddit, on which the 5-block performance of 53.8% narrowly outperforms random performance.

Aggregate model performance drops from 94.5% for block-size 1 to 77.9% for block-size 5, leaving significant room in larger block-size to measure future model improvements.

Although increasing the block size leads to a more challenging task for current models, we argue models should not be evaluated on a single block size, but on several block sizes, with each block size giving a perspective on the model's performance at a specific point between local and global coherence (Van Dijk, 1985).

5 Limitations and Future Work

Shuffling vs. Coherence. In this work, we propose an improved setting for the Shuffle Test, the most popular probe to measure textual coherence. However, many linguistic phenomena necessary for coherence of text cannot be measured by shuffling sentence order. In the long-run, the community should build more elaborate coherence measures, to build a more complete picture of model capabilities and limitations.

[2]https://www.upwork.com

Coherence in Long Text. We limited our analysis to texts with up to 512 words, a common constraint in pre-trained Transformers. Recent progress in model architectures open the possibility to process longer text, with models such as the Reformer (Kitaev et al., 2019), Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020) processing sequences of several thousand words. With longer sequences, one can further increase the block-size of the k-Blocked Shuffle Test (i.e., k=20) to gain understanding of model's ability to discern global coherence (Van Dijk, 1985), or main topics and subtopics (Hearst, 1997).

Specialized Coherence Models. In this work, we limit our analysis to popular models out-of-the-box, establishing baseline performances for the Zero-Shot KBST. Future work should establish whether performance can be further improved, for instance using word-level likelihood signals and surprisal profiles.

6 Conclusion

In this work, we discuss a potential limitation in the framing of the Shuffle Test, the most commonly used task to evaluate models for textual coherence. We show that a RoBERTa model can be finetuned to achieve near-perfect performance without necessarily measuring coherence, and propose a new framework: the Zero-Shot Shuffle Test, in which direct supervision is disallowed. Modern NLP architectures can achieve high performance out-of-the-box in this Zero-Shot setting on a variety of textual domains. We find however that models struggle when we introduce a simple modification, k-Blocking, to the shuffling strategy, with accuracy dropping from around 94% to around 78%. The k-Block Shuffle Test in a Zero-Shot setting is a straightforward, reproducible task that can be used to benchmark future NLP architectures to measure coherence capabilities.

Acknowledgments

We would like to thank Katie Stasaski, Dongyeop Kang, and the ACL reviewers for their helpful comments. This work was supported by a Microsoft BAIR Commons grant as well as a Microsoft Azure Sponsorship.

Ethical Considerations

We present a method to evaluate models on their ability to measure textual coherence. We have ex-

clusively run experiments in the English language, and even though we expect the method to be adaptable to other languages, we have not verified this assumption experimentally and limit our claims to the English language.

For the human evaluation, we paid the annotator above the minimum wage, and do not release any personal identifiable information. We did not collect payment information and relied on a third party (Upwork.com) for remuneration.

References

R. Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34:1–34.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Micha Elsner and Eugene Charniak. 2011. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129, Portland, Oregon, USA. Association for Computational Linguistics.

Marti A. Hearst. 1997. Text Tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2019. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. 2018. Sentence ordering and coherence modeling using recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tasnim Mohiuddin, Prathyusha Jwalapuram, Xiang Lin, and Shafiq Joty. 2020. Coheval: Benchmarking coherence models. *arXiv preprint arXiv:2004.14626*.

Han Cheol Moon, Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A unified neural coherence model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

Dat Tien Nguyen and Shafiq Joty. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1320–1330.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712.

Teun A Van Dijk. 1985. Semantic discourse analysis. *Handbook of discourse analysis*, 2:103–136.

Yuan Wang and Minghe Guo. 2014. A short analysis of discourse coherence. *Journal of Language Teaching and Research*, 5(2):460.

Thomas Wolf, Quentin Lhoest, Patrick von Platen, Yacine Jernite, Mariama Drame, Julien Plu, Julien Chaumond, Clement Delangue, Clara Ma, Abhishek Thakur, Suraj Patil, Joe Davison, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angie McMillan-Major, Simon Brandeis, Sylvain Gugger, François Lagunas, Lysandre Debut, Morgan Funtowicz, Anthony Moi, Sasha Rush, Philipp Schmidd, Pierric Cistac, Victor Muštar, Jeff Boudier, and Anna Tordjmann. 2020. Datasets. *GitHub. Note: https://github.com/huggingface/datasets*, 1.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization

Yixin Liu
Carnegie Mellon University
yixinl2@cs.cmu.edu

Pengfei Liu *
Carnegie Mellon University
pliu3@cs.cmu.edu

Abstract

In this paper, we present a conceptually simple while empirically powerful framework for abstractive summarization, SIMCLS, which can bridge the gap between the *learning objective* and *evaluation metrics* resulting from the currently dominated sequence-to-sequence learning framework by **formulating text generation as a reference-free evaluation problem** (i.e., quality estimation) assisted by *contrastive learning*. Experimental results show that, with minor modification over existing top-scoring systems, SimCLS can improve the performance of existing top-performing models by a large margin. Particularly, 2.51 absolute improvement against BART (Lewis et al., 2020) and 2.50 over PEGASUS (Zhang et al., 2020a) w.r.t ROUGE-1 on the CNN/DailyMail dataset, driving the state-of-the-art performance to a new level. We have open-sourced our codes and results: https://github.com/yixinL7/SimCLS. Results of our proposed models have been deployed into EXPLAINABOARD (Liu et al., 2021a) platform, which allows researchers to understand our systems in a more fine-grained way.

1 Introduction

Sequence-to-sequence (Seq2Seq) neural models (Sutskever et al., 2014) have been widely used for language generation tasks, such as abstractive summarization (Nallapati et al., 2016) and neural machine translation (Wu et al., 2016). While abstractive models (Lewis et al., 2020; Zhang et al., 2020a) have shown promising potentials in the summarization task, they share the widely acknowledged challenges of Seq2Seq model training. Specifically, Seq2Seq models are usually trained under the framework of Maximum Likelihood Estimation (MLE) and in practice they are commonly trained with the *teacher-forcing* (Williams and Zipser, 1989) algorithm. This introduces a gap between the *objective function* and the *evaluation metrics*, as the objective function is based on local, token-level predictions while the evaluation metrics (e.g. ROUGE (Lin, 2004)) would compare the holistic similarity between the gold references and system outputs. Furthermore, during the test stage the model needs to generate outputs autoregressivelly, which means the errors made in the previous steps will accumulate. This gap between the *training* and *test* has been referred to as the *exposure bias* in the previous work (Bengio et al., 2015; Ranzato et al., 2016).

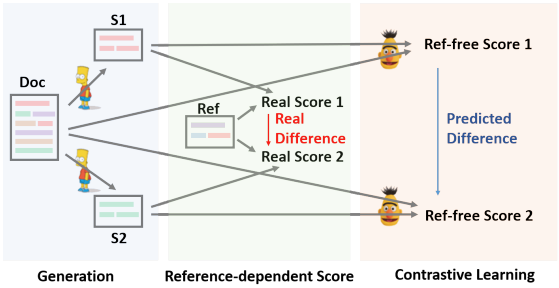


Figure 1: SimCLS framework for two-stage abstractive summarization, where Doc, S, Ref represent the document, generated summary and reference respectively. At the first stage, a Seq2Seq generator (BART) is used to generate candidate summaries. At the second stage, a scoring model (RoBERTa) is used to predict the performance of the candidate summaries based on the source document. The scoring model is trained with contrastive learning, where the training examples are provided by the Seq2Seq model.

A main line of approaches (Paulus et al., 2018; Li et al., 2019) proposes to use the paradigm of Reinforcement Learning (RL) to mitigate the aforementioned gaps. While RL training makes it possible to train the model with rewards based on global predictions and closely related to the evaluation metrics, it introduces the common challenges of deep RL. Specifically, RL-based training suffers from the noise gradient estimation (Greensmith et al., 2004) problem, which often makes the training un-

*Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1065–1072
August 1–6, 2021. ©2021 Association for Computational Linguistics

stable and sensitive to hyper-parameters. Minimum risk training, as an alternative, has also been used in the language generation tasks (Shen et al., 2016; Wieting et al., 2019). However, the accuracy of the estimated loss is restricted by the number of sampled outputs. Other methods (Wiseman and Rush, 2016; Norouzi et al., 2016; Edunov et al., 2018) aim to extend the framework of MLE to incorporate sentence-level scores into the objective functions. While these methods can mitigate the limitations of MLE training, the relation between the evaluation metrics and the objective functions used in their methods can be indirect and implicit.

Among this background, in this work we generalize the paradigm of contrastive learning (Chopra et al., 2005) to introduce an approach for abstractive summarization which achieves the goal of directly optimizing the model with the corresponding evaluation metrics, thereby mitigating the gaps between training and test stages in MLE training. While some related work (Lee et al., 2021; Pan et al., 2021) have proposed to introduce a contrastive loss as an augmentation of MLE training for conditional text generation tasks, we instead choose to disentangle the functions of contrastive loss and MLE loss by introducing them at different stages in our proposed framework.

Specifically, inspired by the recent work of Zhong et al. (2020); Liu et al. (2021b) on text summarization, we propose to use a two-stage model for abstractive summarization, where a Seq2Seq model is first trained to generate candidate summaries with MLE loss, and then a parameterized evaluation model is trained to rank the generated candidates with contrastive learning. By optimizing the generation model and evaluation model at separate stages, we are able to train these two modules with supervised learning, bypassing the challenging and intricate optimization process of the RL-based methods.

Our main contribution in this work is to approach metric-oriented training for abstractive summarization by proposing a generate-then-evaluate two-stage framework with contrastive learning, which not only put the state-of-the-art performance on CNN/DailyMail to a new level (2.2 ROUGE-1 improvement against the baseline model), also demonstrates the great potentials of this two-stage framework, calling for future efforts on optimizing Seq2Seq models using methods beyond maximum likelihood estimation.

2 Contrastive Learning Framework for Abstractive Summarization

Given a source document D and a reference summary $\hat{S}$, the goal of an abstractive summarization model f is to generate the candidate summary $S = f(D)$ such that it receives the highest score $m = M(S, \hat{S})$ assigned by an evaluation metric M. In this work, we break down the holistic generation process into two stages which consist of a *generation model* g for generating candidate summaries and a *evaluation model* h for scoring and selecting the best candidate. Fig 1 illustrates the general framework.

Stage I: Candidate Generation The generation model $g(\cdot)$ is a Seq2Seq model trained to maximize the likelihood of reference summary $\hat{S}$ given the source document D. The pre-trained $g(\cdot)$ is then used to produce multiple candidate summaries $S_1, \cdots, S_n$ with a sampling strategy such as Beam Search, where n is the number of sampled candidates.

Stage II: Reference-free Evaluation The high-level idea is that a better candidate summary S_i should obtain a higher quality score w.r.t the source document D. We approach the above idea by contrastive learning and define an *evaluation function* $h(\cdot)$ that aims to assign different scores $r_1, \cdots, r_n$ to the generated candidates solely based on the similarity between the source document and the candidate S_i, i.e., $r_i = h(S_i, D)$. The final output summary S is the candidate with the highest score:

$$S = \underset{S_i}{\operatorname{argmax}}\, h(S_i, D). \tag{1}$$

Here, we instantiate $h(\cdot)$ as a large pre-trained self-attention model, RoBERTa (Liu et al., 2019). It is used to encode S_i and D separately, and the cosine similarity between the encoding of the first tokens is used as the similarity score r_i.

Contrastive Training Instead of explicitly constructing a positive or negative example as most existing work with contrastive learning have adopted (Chen et al., 2020; Wu et al., 2020), here the "*contrastiveness*" is reflect in the diverse qualities of naturally generated summaries evaluated by a parameterized model $h(\cdot)$. Specifically, we introduce a ranking loss to $h(\cdot)$:

$$\begin{aligned} L = &\sum_i \max(0, h(D, \tilde{S}_i) - h(D, \hat{S})) \\ &+ \sum_i \sum_{j>i} \max(0, h(D, \tilde{S}_j) - h(D, \tilde{S}_i) + \lambda_{ij}), \end{aligned} \tag{2}$$

where $\tilde{S}_1, \cdots, \tilde{S}_n$ is descendingly sorted by $M(\tilde{S}_i, \hat{S})$. Here, $\lambda_{ij} = (j-i)*\lambda$ is the corresponding margin that we defined following Zhong et al. (2020), and λ is a hyper-parameter.[1] M can be any automated evaluation metrics or human judgments and here we use ROUGE (Lin, 2004).

3 Experiments

3.1 Datasets

We use two datasets for our experiments. The dataset statistics are listed in Appendix A.

`CNNDM` CNN/DailyMail[2] (Hermann et al., 2015; Nallapati et al., 2016) dataset is a large scale news articles dataset.

`XSum` XSum[3] (Narayan et al., 2018) dataset is a highly abstractive dataset containing online articles from the British Broadcasting Corporation (BBC).

3.2 Evaluation Metrics

We use ROUGE-1/2/L (R-1/2/L) as the main evaluation metrics for our experiments. We also evaluate our model on the recently developed semantic similarity metrics, namely, BERTScore (Zhang et al., 2020b) and MoverScore (Zhao et al., 2019).

3.3 Base Systems

As the generation model and the evaluation model in our two-stage framework are trained separately, we use pre-trained state-of-the-art abstractive summarization systems as our generation model. Specifically, we use **BART** (Lewis et al., 2020) and **Pegasus** (Zhang et al., 2020a) as they are popular and have been comprehensively evaluated.

3.4 Training Details

For baseline systems, we use the checkpoints provided by the *Transformers*[4] (Wolf et al., 2020) library. We use diverse beam search (Vijayakumar et al., 2016) as the sampling strategy to generate candidate summaries. We use 16 groups for diversity sampling, which results in 16 candidates. To train the evaluation model, we use Adam optimizer (Kingma and Ba, 2015) with learning rate scheduling. The model performance on the validation set is used to select the checkpoint. More details are described in Appendix B.

[1] As it is insensitive, we fix it to 0.01 in our experiments.

[2] https://cs.nyu.edu/~kcho/DMQA/

[3] https://github.com/EdinburghNLP/XSum

[4] https://github.com/huggingface/transformers

System	R-1	R-2	R-L	BS	MS
BART*	44.16	21.28	40.90	-	-
Pegasus*	44.17	21.47	41.11	-	-
Prophet*	44.20	21.17	41.30	-	-
GSum*	45.94	**22.32**	42.48	-	-
Origin	44.39	21.21	41.28	64.67	58.67
Min	33.17	11.67	30.77	58.09	55.75
Max	54.36	28.73	50.77	70.77	61.67
Random	43.98	20.06	40.94	64.65	58.60
SimCLS	**46.67**†	22.15†	**43.54**†	**66.14**†	**59.31**†

Table 1: Results on `CNNDM`. **BS** denotes BERTScore, **MS** denotes MoverScore. **Origin** denotes the original performance of the baseline model. **Min**, **Max**, **Random** are the oracles that select candidates based on their ROUGE scores. †: significantly better than the baseline model (Origin) ($p < 0.01$). *: results reported in the original papers.

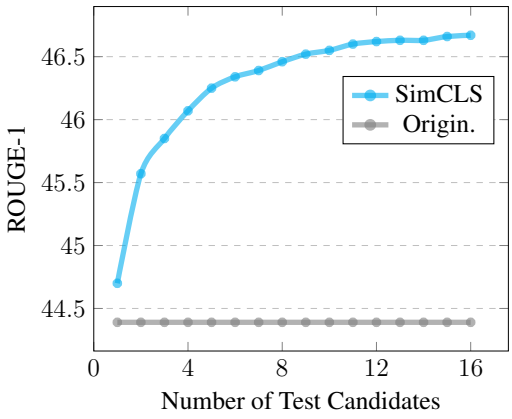


Figure 2: Test performance with different numbers of candidate summaries on `CNNDM`. **Origin** denotes the original performance of the baseline model.

3.5 Results on CNNDM dataset

The results on `CNNDM` dataset are shown in Tab. 1. We use the pretrained BART[5] as the base generation model (**Origin**). We use **BART**, **Pegasus**, **GSum** (Dou et al., 2021) and **ProphetNet** (Qi et al., 2020) for comparison. Notably, the Max oracle which always selects the best candidate has much better performance than the original outputs, suggesting that using a diverse sampling strategy can further exploit the potential power of the pretrained abstractive system. Apart from ROUGE, we also present the evaluation results on semantic similarity metrics. Our method is able to outperform the baseline model on all metrics, demonstrating its improvement is beyond exploiting the potential artifacts of ROUGE. While the scale of improvement is harder to interpret with these metrics, we note that the improvement is able to pass the significance test.

[5] 'facebook/bart-large-cnn'

System	Summary	Article
Ref.	chris ramsey says he has no problem shaking hands with john terry . queens park rangers host chelsea in the premier league on sunday . terry was once banned and fined for racist comments at loftus road . rio ferdinand , brother of anton , will not be fit to play against chelsea .	**queens park rangers manager chris ramsey has revealed he will have no problem shaking john terry's hand in light of the racist comments the former england captain directed at former rs defender anton ferdinand four years ago .** ***terry , who will line up against ramsey's side , was banned for four games and fined # 220,000 for the remarks made in october 2011 during chelsea's 1-0 defeat at loftus road .*** but ramsey , the premier league's only black manager , thinks the issue has been dealt with . ... ' i don't know what his feelings are towards me . as long as there wasn't anything on the field that was unprofessional by him , i would shake his hand . . **queens park rangers manager chris ramsey speaks to the media on friday ahead of the chelsea match .** chelsea captain john terry controls the ball during last weekend's premier league match against stoke . ramsey arrives for friday's pre-match press conference as qpr prepare to host chelsea at loftus road . ' the whole episode for british society sat uncomfortably . it's not something we want to highlight in football . it happened and it's being dealt with . we have to move on . and hopefully everyone has learned something from it . ' . *ramsey revealed that rio ferdinand , who labelled terry an idiot for the abuse aimed at his brother , won't be fit in time for a reunion with the chelsea skipper this weekend .* but the 52-year-old suspects his player's one-time england colleague will be on the receiving end of a hostile welcome from the home fans on his return the scene of the unsavoury incident . ... ferdinand and terry argue during qpr's 1-0 victory against chelsea at loftus road in october 2011 . **rio ferdinand , brother of anton , will not be fit for sunday's match against chelsea .**
SimCLS	queens park rangers host chelsea in the premier league on sunday . qpr boss chris ramsey says he will have no problem shaking john terry's hand . terry was banned for four games and fined # 220,000 for racist comments . rio ferdinand , brother of anton , will not be fit for the match at loftus road .	
Origin.	john terry was banned for four games and fined # 220,000 for the remarks made in october 2011 during chelsea's 1-0 defeat at loftus road . terry will line up against chris ramsey's side on sunday . rio ferdinand , who labelled terry an idiot for the abuse aimed at his brother , won't be fit in time for a reunion with the chelsea skipper this weekend .	

Table 2: Sentence alignments between source articles and summaries on CNNDM dataset. The aligned sentences for reference and our summaries are **bolded** (they are the same in this example). The aligned sentences for baseline summaries are *italicized*. **Origin** denotes the original performance of the baseline model.

Level	System	Precision	Recall	F-Score
Entity	Origin	40.70	59.13	48.22
	SimCLS	**43.36**	**59.79**	**50.27**
Sentence	Origin	38.11	38.65	37.18
	SimCLS	**42.58**	**40.22**	**40.12**

Table 3: Performance analysis on CNNDM dataset. **Origin** denotes the original performance of the baseline model.

With the constraints of computation power, we try to use as many candidates as possible for the evaluation model training. However, we also notice that our method is robust to the specific number of candidates, as during test we found that our model is still able to outperform the baseline model with fewer candidates, which is illustrated in Fig. 2.

3.6 Fine-grained Analysis

To demonstrate that our method is able to make meaningful improvement w.r.t the summary quality, here we compare our method with the baseline model at different semantic levels on CNNDM.

3.6.1 Entity-level

Inspired by the work of Gekhman et al. (2020) and Jain et al. (2020), we compare the model performance w.r.t the *salient entities*, which are entities in source documents that appear in the reference summaries. Specifically, (1) we extract the entities from the source documents,[6] (2) select the *salient entities* based on the entities in reference summaries, (3) compare the *salient entities* with entities in candidate summaries. Results in Tab. 3 demonstrate that our method can better capture the important semantic information of the source documents.

3.6.2 Sentence-level

Sentence Alignments Here we investigate if our method makes sentence-level differences compared to the baseline model. Specifically, (1) we match each sentence in the summaries to a sentence in the source documents based on their similarity (indicated by ROUGE scores),[7] (2) compute the sentence-level similarity between the reference and system-generated summaries based on the overlaps of their matched sentences in the source documents. The results in Tab. 3 demonstrate that the generated summaries of our method is more similar to the reference summaries at the sentence level.

Positional Bias In Tab. 2, we present a case study of the sentence alignment. We use the same matching approach to map the summary sentences to the sentences in source articles. In this example, the output of our method focuses on the same sentences as the reference summary does, while the baseline summary focuses on some different sentences.

Interestingly, the reference summary focuses on the very last sentence in the article, and our method can follow this pattern. Upon examining this pattern, we notice a positional bias of abstractive models when handling long source articles (more than

[6] We use a pre-trained NER model provided by spaCy to extract the entities: https://spacy.io/

[7] Notably, this matching approach formulates an extractive oracle when reference summaries are used for matching, which achieves 54.54/30.73/50.35 ROUGE-1/2/L scores.

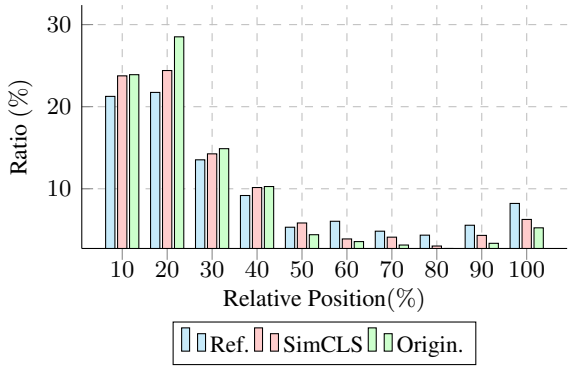


Figure 3: Positional Bias. X-asis: the relative position of the matched sentence in source documents. Y-axis: the ratio of the matched sentences. For fair comparison, articles are first truncated to the generator's maximum input length. **Origin** denotes the original performance of the baseline model.

System	R-1	R-2	R-L	BS	MS
BART*	45.14	22.27	37.25	-	-
Pegasus*	47.21	24.56	39.25	-	-
GSum*	45.40	21.89	36.67	-	-
Origin	47.10	24.53	39.23	69.48	61.34
Min	40.97	19.18	33.68	66.01	59.58
Max	52.45	28.28	43.36	72.56	62.98
Random	46.72	23.64	38.55	69.30	61.23
SimCLS	**47.61**†	**24.57**	**39.44**†	**69.81**†	**61.48**†

Table 4: Results on XSum dataset. **BS** denotes BERTScore, **MS** denotes MoverScore. **Origin** denotes the original performance of the baseline model. **Min**, **Max**, **Random** are the oracles that select candidates based on their ROUGE scores. †: significantly better than the baseline model (Origin) ($p < 0.05$). *: results reported in the original papers.

30 sentences). Fig. 3 shows that the baseline summaries are more likely to focus on the head sentences compared to the references, which may result from the autoregressive generation process of the Seq2Seq models. Our method is able to mitigate this bias, as the candidate sampling process (diverse beam search) generates candidates different from the original outputs, and our evaluation model can assess the holistic quality of the candidates.

3.7 Results on XSum dataset

To evaluate our method's performance beyond CNNDM dataset, we also test our method on XSum dataset, and the results are shown in Tab. 4. Here, we use Pegasus[8] as the base system since it achieves better performance than BART on XSum. We follow the same sampling strategy to generate the training data. However, as this strategy generally results in lower ROUGE-2 score on XSum dataset, we use a different strategy to generate the validation and test data (4 candidates generated by 4 diverse groups). Our method is still able to outperform the baseline, but with a smaller margin compared to CNNDM. Summaries in XSum are shorter (one-sentence) and more abstractive, which restricts the semantic diversity of candidates and makes it harder to make meaningful improvement.

4 Conclusion

In this work, we present a contrastive summarization framework that aims to optimize the quality of generated summaries at summary-level, which mitigates the discrepancy between the training and test stages in the MLE framework. Apart from the significant improvement over the baseline model on CNNDM dataset, we present a comprehensive evaluation at different semantic levels, explaining the sources of the improvement made by our method. Notably, our experimental results also indicate that the existing abstractive systems have the potential of generating candidate summaries much better than the original outputs. Therefore, our work opens up the possibility for future directions including (1) extending this two-stage strategy to other datasets for abstractive models; (2) improving the training algorithms for abstractive models towards a more holistic optimization process.

[8] 'google/pegasus-xsum'

Acknowledgements

We thank Professor Graham Neubig and anonymous reviewers for valuable feedback and helpful suggestions. This work was supported in part by a grant under the Northrop Grumman SOTERIA project and the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks.

In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1171–1179, Cambridge, MA, USA. MIT Press.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Zorik Gekhman, Roee Aharoni, Genady Beryozkin, Markus Freitag, and Wolfgang Macherey. 2020. KoBE: Knowledge-based machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3200–3207, Online. Association for Computational Linguistics.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9).

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6038–6044, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021a. Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yixin Liu, Zi-Yi Dou, and Pengfei Liu. 2021b. RefSum: Refactoring neural summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1448, Online. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Norouzi, Samy Bengio, zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances in Neural Information Processing Systems*, volume 29, pages 1723–1731. Curran Associates, Inc.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hanlu Wu, Tengfei Ma, Lingfei Wu, Tariro Manyumwa, and Shouling Ji. 2020. Unsupervised reference-free summary quality evaluation via contrastive learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3612–3621, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings*

of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Dataset Statistics

Datasets	# Num			Avg. Len	
	Train	Valid	Test	Doc.	Sum.
CNNDM	287K	13K	11K	768.6	55.7
XSum	203K	11K	11K	429.2	23.3

Table 5: Datasets Statistics. Len is the length of tokens.

The source documents and reference summaries are lower-cased. Due to the input length limitation, some source documents are truncated during training.

B Experiment Details

Candidate Generation We use diverse beam search to generate the candidate summaries. We use the same beam search configuration as the original work except those related to diverse beam search. In particular, the diversity penalty is set to 1, and we use 16 diversity groups with 16 beams, which results in 16 candidates.

Model We use the pretrained RoBERTa with 'roberta-base' version provided by the *Transformers* library as our evaluation model, which contains 125M parameters.

Optimizer We use Adam optimizer with learning rate scheduling:

$$lr = 0.002 \cdot \min(\text{step\_num}^{-0.5}, \text{step\_num} \cdot \text{warmup\_steps}^{-1.5}), \quad (3)$$

where the warmup_steps is 10000.

Training details The batch size in our experiments is 32. We evaluate the model performance on the validation set at every 1000 steps, using the averaged ROUGE-1/2/L score as the selecting criteria. The training is converged in 5 epochs, which takes around 40 hours on 4 GTX-1080-Ti GPUs on CNN/DailyMail dataset and 20 hours on XSum dataset.

SaRoCo: Detecting Satire in a Novel Romanian Corpus of News Articles

Ana-Cristina Rogoz, Mihaela Găman, Radu Tudor Ionescu*
University of Bucharest
14 Academiei Street, Bucharest, Romania
*raducu.ionescu@gmail.com

Abstract

In this work, we introduce a corpus for satire detection in Romanian news. We gathered 55,608 public news articles from multiple real and satirical news sources, composing one of the largest corpora for satire detection regardless of language and the only one for the Romanian language. We provide an official split of the text samples, such that training news articles belong to different sources than test news articles, thus ensuring that models do not achieve high performance simply due to overfitting. We conduct experiments with two state-of-the-art deep neural models, resulting in a set of strong baselines for our novel corpus. Our results show that the machine-level accuracy for satire detection in Romanian is quite low (under 73% on the test set) compared to the human-level accuracy (87%), leaving enough room for improvement in future research.

1 Introduction

According to its definition in the Cambridge Dictionary, satire is "a humorous way of criticizing people or ideas"[1]. News satire employs this mechanism in the form of seemingly legitimate journalistic reporting, with the intention of ridiculing public figures, politics or contemporary events (McClennen and Maisel, 2014; Peters and Broersma, 2013; Rubin et al., 2016). Although the articles pertaining to this genre contain fictionalized stories, the intent is not to mislead the public into thinking that the discussed subjects are real. On the contrary, satirical news articles are supposed to reveal their nature by the writing style and comedic devices employed, such as irony, parody or exaggeration. Thus, the intention behind the writing differentiates satirical news (Rubin et al., 2016) from fake news (Meel and Vishwakarma, 2019; Pérez-Rosas et al., 2018; Sharma et al., 2019). However, in some rare cases, the real intent might be deeply buried in the complex irony and subtleties of news satire (Barbieri et al., 2015a), which has the effect of fiction being deemed as factual (Zhang et al., 2020). Even so, there is a clear distinction between satirical and fake news. In fake news, the intent is to deceive the readers in thinking that the news is real, while presenting fake facts to influence the readers' opinion. Since our study is focused on satire detection, we consider discussing research on fake news detection as being out of our scope. At the same time, we acknowledge the growing importance of detecting fake news and the fact that an accurate differentiation of satirical from legitimate journalistic reports might be seen as a starting point in controlling the spread of deceptive news (De Sarkar et al., 2018).

Satire detection is an important task that could be addressed prior to the development of conversational systems and robots that interact with humans. Certainly, the importance of understanding satirical (funny, ridiculous or ironical) text becomes obvious when we consider a scenario in which a robot performs a dangerous action because it takes a satirical comment of the user too literally. Given the relevance of the task for the natural language processing community, satire detection has already been investigated in several well-studied languages such as Arabic (Saadany et al., 2020), English (Burfoot and Baldwin, 2009; De Sarkar et al., 2018; Goldwasser and Zhang, 2016; Yang et al., 2017), French (Ionescu and Chifu, 2021; Liu et al., 2019), German (McHardy et al., 2019), Spanish (Barbieri et al., 2015b) and Turkish (Toçoğlu and Onan, 2019). Through the definition of satire, the satire detection task is tightly connected to irony and sarcasm detection. These tasks strengthen or broaden the language variety with languages such as Ara-

[1] https://dictionary.cambridge.org/dictionary/english/satire

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1073–1079
August 1–6, 2021. ©2021 Association for Computational Linguistics

Data Set	Language	#articles		
		Regular	Satirical	Total
(Burfoot and Baldwin, 2009)	English	4,000	233	4,233
(Frain and Wubben, 2016)	English	1,705	1,706	3,411
(Goldwasser and Zhang, 2016)	English	10,921	1,225	12,146
(Ionescu and Chifu, 2021)	French	5,648	5,922	11,570
(Li et al., 2020)	English	6,000	4,000	10,000
(Liu et al., 2019)	French	2,841	2,841	5,682
(McHardy et al., 2019)	German	320,219	9,643	329,862
(Ravi and Ravi, 2017)	English	1,272	393	1,665
(Saadany et al., 2020)	Arabic	3,185	3,710	6,895
(Toçoğlu and Onan, 2019)	Turkish	1,000	1,000	2,000
(Yang et al., 2017)	English	168,780	16,249	185,029
SaRoCo (ours)	Romanian	27,980	27,628	55,608

Table 1: Number of regular and satirical news articles in existing corpora versus SaRoCo.

Set	Regular		Satirical		Total	
	#articles	#tokens	#articles	#tokens	#articles	#tokens
Training	18,000	8,174,820	17,949	11,147,169	35,949	19,321,989
Validation	4,986	2,707,026	4,878	3,030,055	9,864	5,737,081
Test	4,994	2,124,346	4,801	1,468,199	9,795	3,592,545
Total	27,980	13,006,192	27,628	15,645,423	55,608	28,651,615

Table 2: Number of samples (#articles) and number of tokens (#tokens) for each subset in SaRoCo.

bic (Karoui et al., 2017), Chinese (Jia et al., 2019), Dutch (Liebrecht et al., 2013) and Italian (Giudice, 2018).

In this work, we introduce SaRoCo[2], the **Sa**tire detection **Ro**manian **Co**rpus, which comprises 55,608 news articles collected from various sources. To the best of our knowledge, this is the first and only data set for the study of Romanian satirical news. Furthermore, SaRoCo is also one of the largest data sets for satirical news detection, being surpassed only two corpora, one for English (Yang et al., 2017) and one for German (McHardy et al., 2019). However, our corpus contains the largest collection of satirical news articles (over 27,000). These facts are confirmed by the comparative statistics presented in Table 1.

Along with the novel data set, we include two strong deep learning methods to be used as baselines in future works. The first method is based on low-level features learned by a character-level convolutional neural network (Zhang et al., 2015), while the second method employs high-level semantic features learned by the Romanian version of BERT (Dumitrescu et al., 2020). The gap between the human-level performance and that of the deep learning baselines indicates that there is enough room for improvement left for future studies. We make our corpus and baselines available online for nonprofit educational and research purposes, under an open-source noncommercial license agreement.

[2] https://github.com/MihaelaGaman/SaRoCo

Sample Part	Average #tokens
Title	24.97
Full Articles	515.24

Table 3: Average number of tokens in full news articles and titles from SaRoCo.

2 Corpus

SaRoCo gathers both satirical and non-satirical news from some of the most popular Romanian news websites. The collected news samples were found in the public web domain, i.e. access is provided for free without requiring any subscription to the publication sources. The entire corpus consists of 55,608 samples (27,628 satirical samples and 27,980 non-satirical samples), having more than 28 million tokens in total, as illustrated in Table 2. Each sample is composed of a title (headline), a body and a corresponding label (satirical or non-satirical). As shown in Table 3, an article has around 515.24 tokens on average, with an average of 24.97 tokens for the headline. We underline that the labels are automatically determined, based on the fact that a publication source publishes either regular or satirical news, but not both.

We provide an official split for our corpus, such that all future studies will use the same training, val-

Category	Example	Translation
Regular	*"Tragedie în zi de sărbătoare"*	*"Tragedy during celebration day"*
	"Demisia lui NE NE se amână"	*"NE NE's resignation is post-poned"*
	"Premierul bulgar NE NE are NE"	*"Bulgarian prime-minister NE NE has NE"*
	"A murit actorul NE NE"	*"The actor NE NE died"*
	"Metroul din NE NE se deschide azi"	*"Subway to NE NE opens up today"*
Satirical	*"Comedia cu pălărioară de staniol"*	*"Comedy with little tin-foil hat"*
	"10 restricţii dure pe care NE le pregăteşte pe ascuns"	*"10 harsh restrictions that NE is planning in secrecy"*
	"Câţi pokemoni ai prins azi?"	*"How many pokemons did you catch today?"*
	"Biserica NE lansează apa sfinţită cu aromă"	*"The NE Church launches flavored holy water"*
	"Dragostea în vremea sclerozei"	*"Love in the time of sclerosis"*

Table 4: Examples of news headlines from SaRoCo.

idation and test sets, easing the direct comparison with prior results. Following McHardy et al. (2019), we use disjoint sources for training, validation and test, ensuring that models do not achieve high performance by learning author styles or topic biases particular to certain news websites. While crawling the public news articles, we selected the same topics (culture, economy, politics, social, sports, tech) and the same time frame (between 2011 and 2020) for all news sources to control for potential biases induced by uneven topic or time distributions across the satirical and non-satirical genres.

After crawling satirical and non-satirical news samples, our first aim was to prevent discrimination based on named entities. The satirical character of an article should be inferred from the language use rather than specific clues, such as named entities. For example, certain sources of news satire show preference towards mocking politicians from a specific political party, and an automated system might erroneously label a news article about a member of the respective party as satirical simply based on the presence of the named entity. Furthermore, we even noticed that some Romanian politicians have certain mocking nicknames assigned in satirical news. In order to eliminate named entities, we followed a similar approach as the one used for the MOROCO (Butnaru and Ionescu, 2019) data set. Thus, all the identified named entities are replaced with the special token NE. Besides eliminating named entities, we also substituted all whitespace characters with space and replaced multiple consecutive spaces with a single space. A set of processed satirical and regular headlines are shown in Table 4.

3 Baselines

Fine-tuned Ro-BERT. Our first baseline consists of a fine-tuned Romanian BERT (Dumitrescu et al., 2020), which follows the same transformer-based model architecture as the original BERT (Devlin et al., 2019). According to Dumitrescu et al. (2020), the Romanian BERT (Ro-BERT) attains better results than the multilingual BERT on a range of tasks. We therefore assume that the Romanian BERT should represent a stronger baseline for our Romanian corpus.

We use the Ro-BERT encoder to encode each text sequence into a list of token IDs. The tokens are further processed by the model, obtaining the corresponding 768-dimensional embeddings. At this point, we add a global average pooling layer to obtain a Continuous Bag-of-Words (CBOW) representation for each sequence of text, followed by a Softmax output layer with two neural units, each predicting the probability for one category, either non-satirical or satirical. To obtain the final class label for a text sample, we apply *argmax* on the two probabilities. We fine-tune the whole model for 10 epochs on mini-batches of 32 samples, using the Adam with decoupled weight decay (AdamW) optimizer (Loshchilov and Hutter, 2019), with a learning rate of 10^{-7} and the default value for ϵ.

Character-level CNN. The second baseline model considered in the experiments is a Convolutional Neural Network (CNN) that operates at the character level (Zhang et al., 2015). We set the input size to 1,000 characters. After the input layer, we add an embedding layer to encode each character into a vector of 128 components. The optimal architecture for the task at hand proved to be composed of three convolutional (conv) blocks, each having a conv layer with 64 filters applied at stride 1, followed by Scaled Exponential Linear Unit (SELU) activation. From the first block to the third block, the convolutional kernel sizes are 5, 3 and 1, respectively. Max-pooling with a filter size of 3 is applied after each conv layer. After each conv block, we insert a Squeeze-and-Excitation block

Method	Validation						Test					
	Acc.	Macro F_1	Satirical		Regular		Acc.	Macro F_1	Satirical		Regular	
			Prec.	Rec.	Prec.	Rec.			Prec.	Rec.	Prec.	Rec.
Ro-BERT	0.8241	0.8160	0.9260	0.6991	0.7633	0.9462	0.7300	0.7150	0.8750	0.5250	0.6700	0.9250
Char-CNN	0.7342	0.7475	0.8023	0.6138	0.6928	0.8520	0.6966	0.7109	0.7612	0.5551	0.6606	0.8326

Table 5: Validation and test results of the character-level CNN and the fine-tuned Ro-BERT applied on SaRoCo.

with the reduction ratio set to $r = 64$, following Butnaru and Ionescu (2019). To prevent overfitting, we use batch normalization and Alpha Dropout (Klambauer et al., 2017) with a dropout rate of 0.5. The final prediction layer is composed of two neural units, one for each class (i.e. legitimate and satirical), with Softmax activation. We use the Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer (Dozat, 2016) with a learning rate of $2 \cdot 10^{-4}$, training the network for 50 epochs on mini-batches of 128 samples.

4 Experiments

Evaluation. We conducted binary classification experiments on SaRoCo, predicting if a given piece of text is either satirical or non-satirical. As evaluation metrics, we employ the precision and recall for each of the two classes. We also combine these scores through the macro F_1 and micro F_1 (accuracy) measures.

Results. In Table 5, we present the results of the two baselines on the SaRoCo validation and test sets. We observe that both models tend to have higher precision scores in detecting satire than in detecting regular news. The trade-off between precision and recall is skewed towards higher recall for the non-satirical news class. Since both models share the same behavior, we conjecture that the behavior is rather caused by the particularities of the satire detection task.

Discriminative feature analysis. We analyze the discriminative features learned by the character-level CNN, which is one of the proposed baseline systems for satire detection. We opted for the character-level CNN in favor of the fine-tuned BERT, as the former method allows us to visualize discriminative features using Grad-CAM (Selvaraju et al., 2017), a technique that was initially used to explain decisions of CNNs applied on images. We adapted this technique for the character-level CNN, then extracted and analyzed the most predictive patterns in SaRoCo. The motivation behind this was to validate that the network's decisions are not based on some biases that escaped our data collection and cleaning process.

Category	Example	Translation
Slang	*"cel mai marfă serial din lume"* *"cocalar"*	*"the dopest TV show in the world"* *"douche"*
Insult	*"odiosul primar"* *"bunicuţ retardat"* *"dugongul ăla slinos de la sectorul 4"*	*"the odious mayor"* *"retarded grandpa"* *"that slender dugong in the 4th sector"*
Repetition	*"Mii de gunoaie care lasă gunoaie au remarcat că [...] plajele [...] s-au umplut de gunoaie, lăsate [...] de gunoaiele care au venit înaintea lor"*	*"Thousands of scums who leave garbage noticed that [...] beaches [...] got full of garbage, left behind [...] by the scums who were there before them"*
Exaggeration Exclamation Irony	*"Ne-am săturat!"* *"Ruşine să le fie!"* *"Chiar nu suntem o naţie de hoţi!"*	*"We're sick of it!"* *"Shame on them!"* *"We're totally not a nation of thieves!"*
Popular Saying	*"a sărit calul"* *"a făcut-o de oaie"* *"minte de găină"*	*"went overboard"* *"messed up"* *"bird brain"*

Table 6: Examples of predictive patterns of satire learned by the character-level CNN.

Category	Example	Translation
Stats	*"Importurile au scăzut cu 2.1% [...] pentru o creştere de 0.1% şi prelungirea scăderii de 1.4% din iulie."*	*"Imports decreased by 2.1% [...] for an increase of 0.1% and the prolongation of the decrease of 1.4% since July."*
Legal terms	*"asasinat"* *"l-au denunţat pe autorul atacului"*	*"assasinated"* *"denounced the perpetrator"*
Weather	*"temperatura în timpul nopţii a scăzut"*	*"the temperature has dropped during the night"*
Political terms	*"scrutinul prezidenţial"* *"prefectura informează că"*	*"presidential election"* *"prefecture informs that"*

Table 7: Examples of predictive patterns of legitimate news learned by the character-level CNN.

In Tables 6 and 7, we present a few examples of interesting patterns considered relevant for predicting satire versus regular news, respectively. A broad range of constructions covering a great variety of styles and significant words are underlined

Method	Acc.	Macro F_1	Satirical		Regular	
			Prec.	Rec.	Prec.	Rec.
Ro-BERT	0.6800	0.6750	0.7800	0.5100	0.6350	0.8550
Char-CNN	0.6500	0.6510	0.6389	0.6900	0.6630	0.6100
Humans	0.8735	0.8711	0.9416	0.7970	0.8332	0.9500

Table 8: Averaged performance of ten human annotators versus deep learning baselines on 200 news headlines from SaRoCo.

via Grad-CAM in the satirical news samples. The network seems to pick up obvious clues such as slang, insults and popular sayings rather than more subtle indicatives of satire, including irony or exaggeration. At the same time, for the real news in SaRoCo, there are fewer categories of predictive patterns. In general, the CNN deems formal, standard news expressions as relevant for regular news. These patterns vary across topics and domains. The CNN also finds that the presence of numbers and statistical clues is indicative for non-satirical content, which is consistent with the observations of Yang et al. (2017). Our analysis reveals that the discriminative features are appropriate for satire detection, showing that our corpus is indeed suitable for the considered task.

Deep models versus humans. Given 100 satirical and 100 non-satirical news headlines (titles) randomly sampled from the SaRoCo test set, we asked ten Romanian human annotators to label each sample as satirical or non-satirical. We evaluated the deep learning methods on the same subset of 200 samples, reporting the results in Table 8. First, we observe that humans have a similar bias as the deep learning models. Indeed, for both humans and models, the trade-off between precision and recall is skewed towards higher precision for the satirical class and higher recall for the non-satirical class. We believe this is linked to the way people and machines make a decision. Humans look for patterns of satire in order to label a sample as satire. If a satire-specific pattern is not identified, the respective sample is labeled as regular, increasing the recall for the non-satirical class. Although humans and machine seem to share the same way of thinking, there is a considerable performance gap in satire detection between humans and machines. Indeed, the average accuracy of our ten human annotators is around 87%, while the state-of-the-art deep learning models do not surpass 68% on the same news headlines. Even on full news articles (see Table 5), the models barely reach an accuracy of 73% on the test set. Hence, we conclude there is a significant performance gap between humans and machines, leaving enough room for exploration in future work on Romanian satire detection.

We would like to emphasize that our human evaluation was performed by casual news readers, and the samples were shown after named entity removal, thus having a fair comparison with the AI models. We underline that named entity removal makes the task more challenging, even for humans.

5 Conclusion

In this work, we presented SaRoCo, a novel data set containing satirical and non-satirical news samples. To the best of our knowledge, SaRoCo is the only corpus for Romanian satire detection and one of the largest corpora regardless of language. We trained two state-of-the-art neural models as baselines for future research on our novel corpus. We also compared the performance of the neural models with the averaged performance of ten human annotators, showing that the neural models lag far behind the human-level performance. Our discriminative feature analysis confirms the limitations of state-of-the-art neural models in detecting satire. Although we selected a set of strong models from the recent literature as baselines for SaRoCo, significant future research is necessary to close the gap with respect to the human-level satire detection performance. Designing models to pick up irony or exaggerations could pave the way towards closing this gap in future work.

Acknowledgments

The authors thank reviewers for their useful remarks. This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI – UEFISCDI, project number PN-III-P1-1.1-TE-2019-0235, within PNCDI III. This article has also benefited from the support of the Romanian Young Academy, which is funded by Stiftung Mercator and the Alexander von Humboldt Foundation for the period 2020-2022.

References

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2015a. Do we criticise (and laugh) in the same way? Automatic detection of multi-lingual satirical news in Twitter. In *Proceedings of IJCAI*, pages 1215–1221.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2015b. Is this tweet satirical? A computational approach for satire detection in Spanish. *Procesamiento de Lenguaje Natural*, 55:135–142.

Clint Burfoot and Timothy Baldwin. 2009. Automatic Satire Detection: Are You Having a Laugh? In *Proceedings of ACL-IJCNLP*, pages 161–164.

Andrei Butnaru and Radu Tudor Ionescu. 2019. MOROCO: The Moldavian and Romanian Dialectal Corpus. In *Proceedings of ACL*, pages 688–698.

Sohan De Sarkar, Fan Yang, and Arjun Mukherjee. 2018. Attending Sentences to detect Satirical Fake News. In *Proceedings of COLING*, pages 3371–3380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, pages 4171–4186.

Timothy Dozat. 2016. Incorporating Nesterov Momentum into Adam. In *Proceedings of ICLR Workshops*.

Ştefan Daniel Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of Romanian BERT. In *Findings of EMNLP*, pages 4324–4328.

Alice Frain and Sander Wubben. 2016. SatiricLR: a Language Resource of Satirical News Articles. In *Proceedings of LREC*, pages 4137–4140.

Valentino Giudice. 2018. Aspie96 at IronITA (EVALITA 2018): Irony Detection in Italian Tweets with Character-Level Convolutional RNN. In *Proceedings of EVALITA*, pages 160–165.

Dan Goldwasser and Xiao Zhang. 2016. Understanding Satirical Articles Using Common-Sense. *Transactions of the Association for Computational Linguistics*, 4:537–549.

Radu Tudor Ionescu and Adrian Gabriel Chifu. 2021. Fresada: A french satire data set for cross-domain satire detection. *arXiv preprint arXiv:2104.04828*.

Xiuyi Jia, Zhao Deng, Fan Min, and Dun Liu. 2019. Three-way decisions based feature fusion for Chinese irony detection. *International Journal of Approximate Reasoning*, 113:324–335.

Jihen Karoui, Farah Zitoune, and Véronique Moriceau. 2017. SOUKHRIA: Towards an Irony Detection System for Arabic in Social Media. In *Proceedings of ACLing*, volume 117, pages 161–168.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-Normalizing Neural Networks. In *Proceedings of NIPS*, pages 972–981.

Lily Li, Or Levi, Pedram Hosseini, and David Broniatowski. 2020. A Multi-Modal Method for Satire Detection using Textual and Visual Cues. In *Proceedings of NLP4IF*, pages 33–38.

Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of WASSA*, pages 29–37.

Zhan Liu, Shaban Shabani, Nicole Glassey Balet, and Maria Sokhn. 2019. Detection of Satiric News on Social Media: Analysis of the Phenomenon with a French Dataset. In *Proceedings of ICCCN*, pages 1–6.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of ICLR*.

Sophia A. McClennen and Remy M. Maisel. 2014. I'm Not Laughing at You, I'm Laughing With You: How to Stop Worrying and Love the Laughter. In *Is Satire Saving Our Nation? Mockery and American Politics*, pages 189–201. Springer.

Robert McHardy, Heike Adel, and Roman Klinger. 2019. Adversarial Training for Satire Detection: Controlling for Confounding Variables. In *Proceedings of NAACL*, pages 660–665.

Priyanka Meel and Dinesh Kumar Vishwakarma. 2019. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 153:112986.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proceedings of COLING*, pages 3391–3401.

Chris Peters and Marcel Jeroen Broersma. 2013. *Rethinking Journalism: Trust and Participation in a Transformed News Landscape*. Routledge.

Kumar Ravi and Vadlamani Ravi. 2017. A novel automatic satire and irony detection using ensembled feature selection and data mining. *Knowledge-Based Systems*, 120:15–33.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. In *Proceedings of CADD*, pages 7–17.

Hadeel Saadany, Constantin Orasan, and Emad Mohamed. 2020. Fake or Real? A Study of Arabic Satirical Fake News. In *Proceedings of RDSM*, pages 70–80.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *Proceedings of ICCV*, pages 618–626.

Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology*, 10(3):1–42.

Mansur Alp Toçoğlu and Aytuğ Onan. 2019. Satire Detection in Turkish News Articles: A Machine Learning Approach. In *Proceedings of Innovate-Data*, pages 107–117.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical News Detection and Analysis using Attention Mechanism and Linguistic Features. In *Proceedings EMNLP*, pages 1979–1989.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Proceedings of NIPS*, pages 649–657.

Yigeng Zhang, Fan Yang, Eduard Constantin Dragut, and Arjun Mukherjee. 2020. Birds of a Feather Flock Together: Satirical News Detection via Language Model Differentiation. *arXiv preprint arXiv:2007.02164*.

Bringing Structure into Summaries: a Faceted Summarization Dataset for Long Scientific Documents

Rui Meng[♣] **Khushboo Thaker**[♣] **Lei Zhang**[♣] **Yue Dong**[♢]
Xingdi Yuan[♠] **Tong Wang**[♠] **Daqing He**[♣]
[♣]School of Computing and Information, University of Pittsburgh
[♢] Mila / McGill University
[♠]Microsoft Research, Montréal
{rui.meng, k.thaker, lez39, dah44}@pitt.edu
yue.dong2@mail.mcgill.ca
{eric.yuan, tong.wang}@microsoft.com

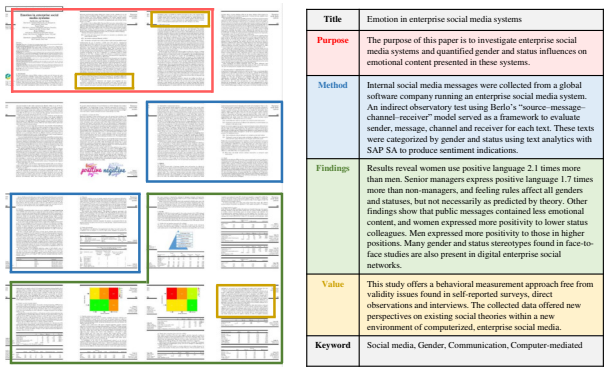
Abstract

Faceted summarization provides briefings of a document from different perspectives. Readers can quickly comprehend the main points of a long document with the help of a structured outline. However, little research has been conducted on this subject, partially due to the lack of large-scale faceted summarization datasets. In this study, we present FacetSum, a faceted summarization benchmark built on Emerald journal articles, covering a diverse range of domains. Different from traditional document-summary pairs, FacetSum provides *multiple summaries*, each targeted at specific sections of a long document, including the purpose, method, findings, and value. Analyses and empirical results on our dataset reveal the importance of bringing structure into summaries. We believe FacetSum will spur further advances in summarization research and foster the development of NLP systems that can leverage the structured information in both long texts and summaries.

1 Introduction

Text summarization is the task of condensing a long piece of text into a short summary without losing salient information. Research has shown that a well-structured summary can effectively facilitate comprehension (Hartley et al., 1996; Hartley and Sydes, 1997). A case in point is the *structured abstract*, which consists of multiple segments, each focusing on a specific facet of a scientific publication (Hartley, 2014), such as background, method, conclusions, etc. The structure therein can provide much additional clarity for improved comprehension and has long been adopted by databases and publishers such as MEDLINE and Emerald.

Despite these evident benefits of structure, summaries are often framed as a linear, structure-less sequence of sentences in the flourishing array of summarization studies (Nallapati et al., 2017; See et al., 2017; Paulus et al., 2018; Grusky et al., 2018; Narayan et al., 2018; Sharma et al., 2019; Lu et al., 2020; Cachola et al., 2020). We postulate that a primary reason for this absence of structure lies in the lack of a high-quality, large-scale dataset with structured summaries. In fact, existing studies in faceted summarization (Huang et al., 2020; Tauchmann et al., 2018; Jaidka et al., 2016; Contractor et al., 2012; Kim et al., 2011; Jaidka et al., 2018; Stead et al., 2019) are often conducted with rather limited amount of data that are grossly insufficient to meet today's ever-growing model capacity.



Title	Emotion in enterprise social media systems
Purpose	The purpose of this paper is to investigate enterprise social media systems and quantified gender and status influences on emotional content presented in these systems.
Method	Internal social media messages were collected from a global software company running an enterprise social media system. An indirect observatory test using Berlo's "source–message–channel–receiver" model served as a framework to evaluate sender, message, channel and receiver for each text. These texts were categorized by gender and status using text analytics with SAP SA to produce sentiment indications.
Findings	Results reveal women use positive language 2.1 times more than men. Senior managers express positive language 1.7 times more than non-managers, and feeling rules affect all genders and statuses, but not necessarily as predicted by theory. Other findings show that public messages contained less emotional content, and women expressed more positivity to lower status colleagues. Men expressed more positivity to those in higher positions. Many gender and status stereotypes found in face-to-face studies are also present in digital enterprise social networks.
Value	This study offers a behavioral measurement approach free from validity issues found in self-reported surveys, direct observations and interviews. The collected data offered new perspectives on existing social theories within a new environment of computerized, enterprise social media.
Keyword	Social media, Gender, Communication, Computer-mediated

Figure 1: An example of the proposed FacetSum dataset. Each facet of the structured abstract summarizes different sections of the paper.

We aim to address this issue by proposing the FacetSum dataset. It consists of 60,024 scientific articles collected from Emerald journals, each associated with a *structured* abstract that summarizes the article from distinct aspects including purpose, method, findings, and value. Scale-wise, we empirically show that the dataset is sufficient for training large-scale neural generation models such as BART (Lewis et al., 2020) for adequate generalization. In terms of quality, each structured abstract in FacetSum is provided by the original author(s) of the article, who are arguably in the best position to summarize their own work. We also provide

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1080–1089
August 1–6, 2021. ©2021 Association for Computational Linguistics

# documents					
Train: 46,289 / Dev: 6,000 / Test: 6,000 / OA-Test: 2,243					
# words in abstracts					
	Full	Purpose	Method	Findings	Value
mean	290.4	54.1	52.0	68.6	47.3
std	±82.8	±28.4	±27.8	±32.4	±24.2
# words in paper sections					
	Full	Intro.	Method	Result	Conc.
recall%	-	84.3%	67.0%	72.4%	79.0%
mean	6,827	885	1,194	2,371	747
std	±2,704	±557	±861	±1,466	±567

Table 1: Statistics of the FacetSum dataset.

quantitative analyses and baseline performances on the dataset with mainstream models in Sections 2 and 3.

2 FacetSum for Faceted Summarization

The FacetSum dataset is sourced from journal articles published by Emerald Publishing[1] (Figure 1). Unlike many publishers, Emerald imposes explicit requirements that authors summarize their work from multiple aspects (Emerald, 2021): **Purpose** describes the motivation, objective, and relevance of the research; **Method** enumerates specific measures taken to reach the objective, such as experiment design, tools, methods, protocols, and datasets used in the study; **Findings** present major results such as answers to the research questions and confirmation of hypotheses; and **Value** highlights the work's value and originality[2]. Together, these facets give rise to a comprehensive and informative structure in the abstracts of the Emerald articles, and by extension, to FacetSum's unique ability to support faceted summarization.

2.1 General Statistics

We collect 60,532 publications from Emerald Publishing spanning 25 domains. Table 1 lists some descriptive statistics of the dataset. Since FacetSum is sourced from journal articles, texts therein are naturally expected to be longer compared to other formats of scientific publications. In addition, although each facet is more succinct than the traditional, structure-less abstracts, a full length abstract containing all facets can be considerably longer. Empirically, we compare the source and the target lengths with some existing summarization datasets in similar domains including CLPubSum (Collins et al., 2017), PubMed (Cohan et al., 2018), ArXiv (Cohan et al., 2018), SciSummNet (Yasunaga et al., 2019), and SciTldr (Cachola et al., 2020). On average, the source length in FacetSum is 58.9% longer (6,827 vs 4,297), and the target length is 37.0% longer (290.4 vs 212.0).

From a summarization perspective, these differences imply that FacetSum may pose significantly increased modeling and computation challenges due to the increased lengths in both the source and the target. Moreover, the wide range of research domains (Figure 3, Appendix D) may also introduce much linguistic diversity w.r.t. vocabulary, style, and discourse. Therefore, compared to existing scientific publication datasets that only focus on specific academic disciplines (Cohan et al., 2018; Cachola et al., 2020), FacetSum can also be used to assess a model's robustness in domain shift and systematic generalization.

To facilitate assessment of generalization, we reserve a dev and a test set each consisting of 6,000 randomly sampled data points; the remaining data are intended as the training set. We ensure that the domain distribution is consistent across all three subsets. Besides, we intentionally leave out Open-Access papers as another test set, to facilitate researchers who do not have full Emerald access[3].

2.2 Structural Alignment

In this section, we focus our analysis on one of the defining features of FacetSum — its potential to support faceted summarization. Specifically, we investigate how the abstract structure (i.e., facets) aligns with the article structure. Given an abstract facet A and its corresponding article S, we quantify this alignment by:

$$S_A = \{\arg\max_{s_i \in S}(\texttt{Rouge-1}(s_i, a_j)) : a_j \in A\} \quad (1)$$

Semantically, S_A consists of sentence indices in S that best align with each sentence in A.

Sentence-level Alignment We first plot the tuples $\{(s_i, i/|S|) : i \in S_A\}$, where s_i is the i-th sentence in S, and $|S|$ is the number of sentences in S. Intuitively, the plot density around position $i/|S|$ entails the degree of alignment between the facet

[1]The data has been licensed to researchers at subscribing institutions to use (including data mining) for non-commercial purposes. See detailed policies at https://www.emerald.com/

[2]There are three optional facets (about research, practical and social implications) that are missing from a large number of articles and hence omitted in this study.

[3]Both the split information of FacetSum and the code for scraping and parsing the data are available at https://github.com/hfthair/emerald_crawler

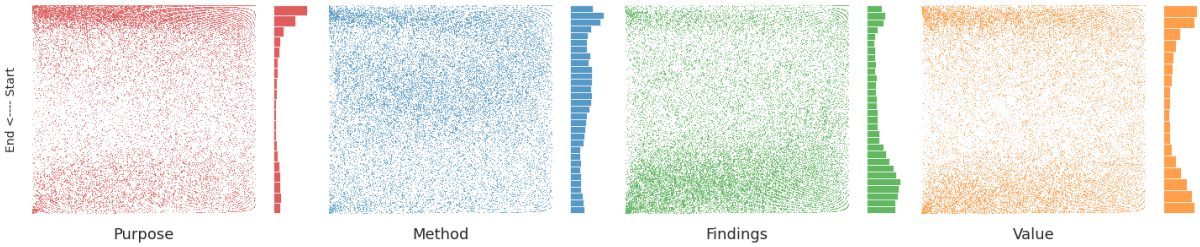


Figure 2: Oracle sentence distribution over a paper. X-axis: 10,000 papers sampled from FacetSum, sorted by full text length from long to short; y-axis: normalized position in a paper. We provide each sub-figure's density histogram on their right.

Abstract \ Paper Section	Full	I+C	Introduction	Method	Result	Conclusion
Full	62.09	56.47	48.47	43.32	49.73	50.42
Purpose	49.76	47.06	44.23	30.12	33.87	36.23
Method	45.36	34.23	30.82	37.53	29.07	28.46
Findings	52.09	45.28	33.65	29.49	42.80	42.35
Value	45.98	42.37	35.29	26.68	32.52	36.85

Table 2: Scores of sentence aligning in Rouge-L.

A and the article S at that position[4]. With 10,000 articles randomly sampled from FacetSum, Figure 2 exhibits distinct differences in the density distribution among the facets in FacetSum. For example, with $A =$ Purpose, resemblance is clearly skewed towards the beginning of the articles, while Findings are mostly positioned towards the end; the Method distribution is noticeably more uniform than the others. These patterns align well with intuition, and are further exemplified by the accompanying density histograms.

Section-level Alignment We now demonstrate how different abstract facets align with different sections in an article. Following conventional structure of scientific publications (Suppe, 1998; Rosenfeldt et al., 2000), we first classify sections into *Introduction*, *Method*, *Result* and *Conclusion* using keyword matching in the section titles.[5]

Given a section $S^i \subseteq S$ and an abstract $A_j \subseteq A$, we define the section-level alignment $g(S^i, A_j)$ as $\texttt{Rouge-1}(\texttt{cat}(S^i_{A_j}), \texttt{cat}(A_j))$, where $\texttt{cat}(\cdot)$ denotes sentences concatenation, and $S^i_{A_j}$ is defined by Equation (1). Table 2 is populated by varying A_j and S^i across the rows and columns, respectively. **Full** denotes the full paper or abstract (concatenation of all facets). We also include the concatenation of introduction and conclusion (denoted I+C) as a possible value for S^i, due to its demonstrated effectiveness as summaries in prior work (Cachola et al., 2020).

The larger numbers on the diagonal (in red) empirically confirm a strong alignment between FacetSum facets and their sectional counterparts in articles. We also observe a significant performance gap between using I+C and the full paper as S^i. One possible reason is that the summaries in FacetSum (particularly Method and Findings) may contain more detailed information beyond introduction and conclusion. This suggests that for some facets in FacetSum, simple tricks to condense full articles do not always work; models need to instead comprehend and retrieve relevant texts from full articles in a more sophisticated manner.

3 Experiments and Results

We use FacetSum to benchmark a variety of summarization models from state-of-the-art supervised models to unsupervised and heuristics-based models. We also provide the scores of a sentence-level extractive oracle system (Nallapati et al., 2017). We report Rouge-L in this section and include Rouge-1/2 results in Appendix E.

Unsupervised Models vs Heuristics We report performances of unsupervised and heuristics summarization methods (see Table 3). Tailoring to the unique task of generating summaries for a specific facet, we only use the section (defined in Section 2.2) corresponding to a facet as model input. Evaluation is also performed on the concatenation

[4]We use the relative position $i/|S|$ so that all positions are commensurate across multiple documents.

[5]To ensure close-to-perfect precision, we choose keywords that are as specific and prototypical to each section as possible (listed in Appendix A). The resulting recall is around 0.7, i.e. about 70% of sections can be correctly retrieved with the title-keyword matching method. And we find 2,751 (out of 6,000) test samples that all four sections are matched successfully. Though far from perfect, we believe this size is sufficient for the significance of subsequent analyses.

	Model	Source Text	Full	Purpose	Method	Findings	Value
FacetSum Test							
Oracle	Greedy Extractive (Nallapati et al., 2017)	corresponding	60.39	44.66	41.00	46.44	38.10
Heuristic Models	Lead-K	corresponding	36.78	17.83	15.29	15.92	16.08
	Tail-K	sections	33.31	21.67	12.62	16.66	17.43
Unsupervised Models	SumBasic (Vanderwende et al., 2007)	corresponding sections	38.71	18.17	15.41	16.31	16.57
	LexRank (Erkan and Radev, 2004)		42.18	18.72	16.23	18.11	17.75
	LSA (Gong and Liu, 2001)		35.98	18.29	15.86	16.92	16.62
	TextRank (Mihalcea and Tarau, 2004)		41.87	21.67	13.62	18.63	19.23
	HipoRank (Dong et al., 2020)		42.89	22.73	15.20	18.38	19.68
Supervised Models	BART (Lewis et al., 2020)	I+C	44.36	41.14	20.75	14.72	5.85
	BART-Facet	I+C	**47.09**	**43.47**	**29.07**	**30.97**	**28.90**
	BART	full paper	42.74	41.21	20.53	14.33	5.07
	BART-Facet	full paper	45.76	42.55	28.07	28.98	28.70
FacetSum OA-Test							
	BART	I+C	44.97	43.51	26.73	11.79	0.31
	BART-Facet	I+C	51.32	43.66	30.16	32.22	29.68

Table 3: Model performance on FacetSum (Rouge-L). See Table 6 and 7 in Appendix E for full results. **Bold** text indicates the best scores on FacetSum test split in each column.

of all facets (column **Full**), which resembles the traditional research abstract. **Lead-K/Tail-K** are two heuristic-based models that extract the first/last k sentences from the source text.

We observe that heuristic models do not perform well on **Full**, where the unsupervised models can achieve decent performance. Nevertheless, all models perform poorly on summarizing individual facets, and unsupervised models fail to perform better than simple heuristics consistently. The inductive biases of those models may not be good indicators of summary sentences on specific facets. A possible reason is that they are good at locating overall important sentences of a document, but they cannot differentiate sentences of each facet, even we try to alleviate this by using the corresponding section as input.

Supervised Models As for the supervised baseline, we adopt the BART model (Lewis et al., 2020), which has recently achieved SOTA performance on abstractive summarization tasks with scientific articles (Cachola et al., 2020). We propose two training strategies for the BART model, adapting it to handle the unique challenge of faceted summarization in FacetSum. In **BART**, we train the model to generate the concatenation of all facets, joined by special tokens that indicate the start of a specific facet (e.g., |PURPOSE| to indicate the start of Purpose summary). During evaluation, the generated text is split into multiple facets based on the special tokens, and each facet is compared against the corresponding ground-truth summary. In **BART-Facet**, we train the model to generate one specific facet given the source text and an indicator specifies which facet to generate. Inspired by CATTS (Cachola et al., 2020), we prepend section tags at the beginning of each training input to generate summaries for a particular facet (see implementation details in Appendix C).

Empirically, supervised models outperform unsupervised baselines by a large margin (Table 3). Comparing between the two training strategies, BART-Facet outperforms BART significantly. While BART performs comparably on Purpose, performance decreases drastically for subsequent facets, possibly due to current models' inadequacy with long targets. Thus it can perform decently at the beginning of generation (≈40 on Purpose), where the dependency is relatively easy-to-handle. However, the output quality degrades quickly towards the end (≈5 on Value).

With I+C as source text, both training strategies exhibit much better results than using full paper. This is opposite to the observation in Table 2, potentially due to the limitation of the current NLG systems, i.e., the length of source text has crucial impacts to the model performance. With the much extended positional embeddings in our models (10,000 tokens), we suspect some other issues such as long term dependencies may lead to this discrepancy, which warrants further investigation.

4 Conclusion & Future Work

We introduce FacetSum to support the research of faceted summarization, which targets summarizing scientific documents from multiple facets. We provide extensive analyses and results to investigate the characteristics of FacetSum. Our observations call for the development of models capable of handling very long documents and outputting controlled text. Specifically, we will consider exploring the following topics in future work: (1) incorporating methods for long-document processing, such as reducing input length by extracting key sentences (Pilault et al., 2020) or segments (Zhao et al., 2020); (2) examining the possibility of building a benchmark for systematic generalization (Bahdanau et al., 2018) with the domain categories; (3) automatically structuring traditional abstracts (Huang et al., 2020) with FacetSum.

References

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2018. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*.

Mišo Belica. 2021. sumy: Automatic text summarizer. https://github.com/miso-belica/sumy.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. 2020. Tldr: Extreme summarization of scientific documents. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4766–4777.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada. Association for Computational Linguistics.

Danish Contractor, Yufan Guo, and Anna Korhonen. 2012. Using argumentative zones for extractive summarization of scientific articles. In *Proceedings of COLING 2012*, pages 663–678.

Yue Dong, Andrei Romascanu, and Jackie CK Cheung. 2020. Hiporank: Incorporating hierarchical and positional information into graph-based unsupervised long document extractive summarization. *arXiv preprint arXiv:2005.00513*.

Emerald. 2021. Writing an article abstract. https://www.emeraldgrouppublishing.com/how-to/authoring-editing-reviewing/write-article-abstract. [Online; accessed 26-January-2021].

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719.

James Hartley. 2014. Current findings from research on structured abstracts: an update. *Journal of the Medical Library Association: JMLA*, 102(3):146.

James Hartley and Matthew Sydes. 1997. Are structured abstracts easier to read than traditional ones? *Journal of Research in Reading*, 20(2):122–136.

James Hartley, Matthew Sydes, and Anthony Blurton. 1996. Obtaining information accurately and quickly: are structured abstracts more efficient? *Journal of information science*, 22(5):349–356.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Ting-Hao Huang, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Yen-Chia Hsu, and C Lee Giles. 2020. Coda-19: Using a non-expert crowd to annotate research aspects on 10,000+ abstracts in the covid-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2016. Overview of the cl-scisumm 2016 shared task. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, pages 93–102.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2018. Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. *International Journal on Digital Libraries*, 19(2-3):163–171.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, volume 12, pages 1–10. BioMed Central.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3075–3081.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Franklin L Rosenfeldt, John T Dowling, Salvatore Pepe, and Meryl J Fullerton. 2000. How to write a paper for publication. *Heart, Lung and Circulation*, 9(2):82–87.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213.

Connor Stead, Stephen Smith, Peter Busch, and Savanid Vatanasakdakul. 2019. Emerald 110k: a multidisciplinary dataset for abstract sentence classification. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 120–125.

Frederick Suppe. 1998. The structure of a scientific paper. *Philosophy of Science*, 65(3):381–405.

Christopher Tauchmann, Thomas Arnold, Andreas Hanselowski, Christian M Meyer, and Margot Mieskes. 2018. Beyond generic summarization: A multi-faceted hierarchical summarization corpus of large heterogeneous data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7386–7393.

Yao Zhao, Mohammad Saleh, and Peter J Liu. 2020. Seal: Segment-wise extractive-abstractive long-form text summarization. *arXiv preprint arXiv:2006.10213*.

A Keyword List for Identifying Paper Sections

Category	Keyword
Introduction	intro, purpose
Method	design, method, approach
Result	result, find, discuss, analy
Conclusion	conclu, future

Table 4: Keywords for identifying paper sections used in Section 2.2.

B Most Frequent Words in Each Abstract Facet

Facet	Verb	Noun	Adjective
Purpose	aim	paper	social
	examin	purpos	new
	investig	studi	organiz
	explor	manag	differ
	develop	research	public
Method	base	studi	structur
	conduct	data	qualit
	collect	analysi	differ
	test	model	empir
	develop	paper	social
Findings	found	result	signific
	indic	studi	posit
	suggest	manag	social
	provid	effect	differ
	identifi	relationship	higher
Value	provid	studi	new
	contribut	paper	social
	develop	research	differ
	base	manag	empir
	examin	literatur	import

Table 5: Top five frequent verbs/nouns/adjectives in each facet of structured abstract. We preprocess the text with lowercasing, stemming and stopword removal and extract part-of-speech tags using Spacy (Honnibal et al., 2020).

C Implementation Details

To make BART take full text as input, we extend the positional embedding to 10,000 tokens. This was required to leverage long text of papers in `FacetSum` with average length of 6000 words.

Experiments of unsupervised baselines are implemented with Sumy (Belica, 2021) and official code of HipoRank. We tune the hyperparameters of HipoRank with the validation set. The BART experiments are finetuned using Fairseq (Ott et al., 2019), with learning rate of $3e^{-5}$, batch size of 1, max tokens per batch of 10,000 and update frequency of 4. We finetune all models for 20,000 steps with single NVIDIA Tesla V100 16GB and we report the results of the last checkpoint. The small batch size is the consequence of the large input size. For inference, we use beam size of 4 and maximum length of 500/200 tokens for BART/BART-Facet respectively.

D Domains Covered by `FacetSum`

In Figure 3, we show the distribution of domain categories in `FacetSum`.

E Full Results

In this section, we provide additional experiment results. In Table 6, we show the full results of the extractive oracle system (first row in Table 3). In Table 7, we provide full results of all other models (heuristic models, unsupervised models, and supervised models in Table 3).

F Example of Outputs by BART and BART-Facet

In Table 8, we show an example of the generated faceted summaries by BART and BART-Facet of the same paper, compared against the ground-truth faceted abstract.

R1/R2/RL	Full	Purpose	Method	Findings	Value
Full_{body}	64.92/33.75/60.39	57.35/30.24/49.42	53.30/26.40/45.58	59.30/33.25/52.42	53.39/26.84/45.55
IC_{body}	58.82/28.42/54.17	53.60/27.13/45.73	43.13/17.08/35.64	52.03/25.90/44.86	48.97/22.84/41.09
Intro_{body}	53.32/22.96/48.59	**52.51/26.48/44.66**	41.27/16.05/34.03	44.67/17.49/37.10	44.65/17.80/36.47
Method_{body}	52.05/20.52/47.35	45.16/16.61/36.84	**48.60/21.67/41.00**	44.77/17.69/37.67	40.94/13.55/32.94
Result_{body}	**56.85/23.79/51.97**	47.90/18.07/38.96	42.31/14.46/34.41	**53.71/26.32/46.44**	44.93/16.91/36.66
Conclu_{body}	55.26/25.26/50.58	47.76/18.88/38.94	40.53/13.84/32.83	51.81/25.81/44.73	**46.14/19.66/38.10**

Table 6: Full results (Rouge-1/2/L) of the extractive oracle system (Nallapati et al., 2017) on FacetSum. **Bold** text indicates the best scores in the lower four rows in each column.

R1/R2/RL	Full	Purpose	Method	Findings	Value
FacetSum Test					
Lead-K	39.65/11.01/36.78	21.95/4.89/17.83	18.69/5.94/15.29	18.84/4.31/15.92	20.14/3.05/16.08
Tail-K	35.90/10.96/33.31	25.48/7.23/21.67	14.88/2.64/12.62	19.25/4.41/16.66	20.90/4.71/17.43
SumBasic	42.11/10.01/38.71	22.23/4.68/18.17	18.40/5.02/15.41	19.15/3.93/16.31	20.64/3.08/16.57
LexRank	46.35/15.12/42.18	22.97/5.28/18.72	19.44/5.84/16.23	21.66/5.66/18.11	22.39/4.05/17.75
LSA	39.84/9.59/35.98	22.47/4.91/18.29	19.10/5.58/15.86	20.29/4.59/16.92	20.96/3.31/16.62
TextRank	46.90/16.04/41.87	28.29/9.39/21.67	17.55/4.32/13.62	23.90/7.17/18.63	25.99/7.07/19.23
HipoRank	46.48/15.42/42.89	27.71/8.29/22.73	18.27/4.65/15.20	21.75/5.31/18.38	24.54/5.26/19.68
BART I+C	47.21/19.59/44.36	46.61/27.10/41.14	23.85/7.98/20.75	16.84/5.34/14.72	7.21/1.93/5.85
BART-Facet I+C	**50.62/20.97/47.09**	**49.59/28.70/43.47**	**34.61/11.82/29.07**	**36.42/12.63/30.97**	**35.37/11.75/28.90**
BART full body	45.49/18.10/42.74	46.74/27.09/41.21	23.66/7.92/20.53	16.39/4.63/14.33	6.30/1.62/5.07
BART-Facet full body	49.29/19.60/45.76	48.65/27.72/42.55	33.49/11.01/28.07	34.46/10.49/28.98	35.27/11.44/28.70
FacetSum OA-Test					
BART I+C	48.85/20.84/44.97	49.43/29.44/43.51	31.1/10.16/26.73	13.78/4.45/11.79	0.4/0.1/0.31
BART-Facet I+C	48.31/22.63/51.32	49.59/28.69/43.66	35.82/12.84/30.16	37.46/14.02/32.22	35.9/12.75/29.68

Table 7: Full results (Rouge-1/2/L) of different models on FacetSum. **Bold** text indicates the best scores on FacetSum test split in each column.

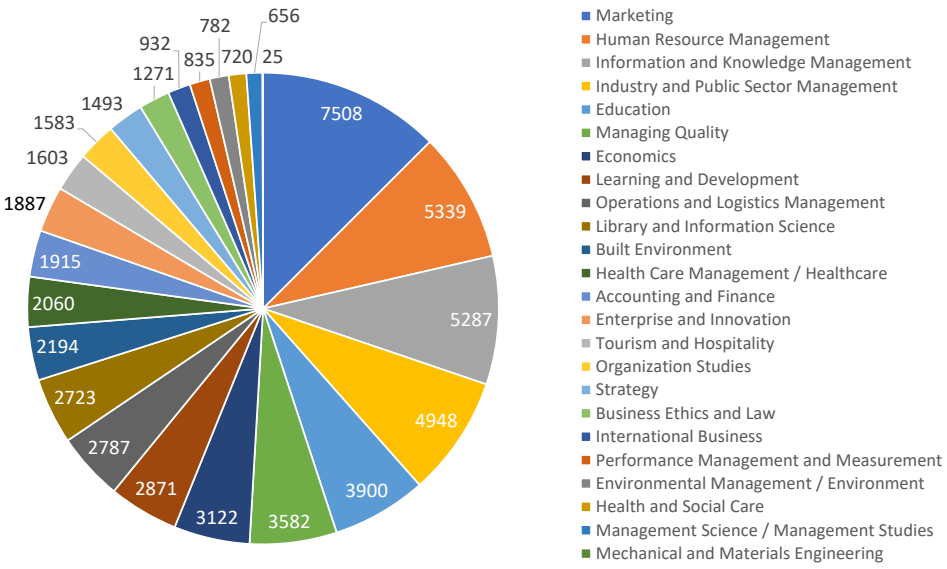


Figure 3: Data distribution of domain categories, sorted in descending order.

Facet	Ground Truth	BART	BART-Facet
Purpose	The purpose of this paper is to advance the information society research by examining and better understanding the impact of the adoption information and communication technologies (ICT) within households on improving sustainability.	The purpose of this paper is to answer the question: Does the adoption of information and communication technologies (ICT) within households contribute to sustainability? For this purpose, the study focuses on addressing the following specific questions: What criteria clearly and deeply describe the ICT adoption within households? What are the criteria that clearly describe sustainability within households, and are there statistically significant correlations between ICT and sustainability?	The purpose of this paper is to answer the question: Does the adoption of information and communication technologies (ICT) within households contribute to sustainability?
Method	The survey questionnaire was used and data collected from 679 Polish households were statistically analyzed to understand the phenomena of ICT adoption and sustainability as well as identify correlations between them.	A survey was carried out in Poland. The data were collected through a structured questionnaire and analyzed using structural equation modeling (SEM).	The study is based on a quantitative research design. The data were collected through a questionnaire survey among the households in Poland. The survey was carried out in two phases. In the first phase, the ICT adoption within households was measured using a structured questionnaire, and in the second phase the data were analyzed using structural equation modeling (SEM) to test the hypothesized relationships.
Findings	The research findings reveal that the ICT adoption is well described by the ICT outlay, information culture, ICT management and ICT quality, whereas sustainability is composed of ecological, economic, socio-cultural and political sustainability. Furthermore, the ICT quality, ICT management and information culture have a significant impact on sustainability, whereas the ICT outlay does not have such an impact.	Results show that ICT is a key enabler of sustainability in households. The results also show that there are statistically significant correlation between the IIT adoption within the households and sustainability.	The results show that the adoption of ICT within households is positively related to sustainability. The results also show that there are statistically significant correlations between the ICT adoption within households and sustainability.
Value	The paper provides and verifies a new theoretical model of sustainable information society to depict various dimensions shaping the ICT adoption and their impact on different types of sustainability in the context of households.	This study is the first to empirically investigate the impact of ICT on sustainability. The findings of this study will be complementary with findings concerning the contribution of IIT to sustainability in enterprises and allow for the advancement in the sustainable information society (SIS) research.	This study contributes to the literature by providing a deeper understanding of the ICT adoption within households and the contribution of ICT to sustainability in transition economies, i.e. the former European Eastern Bloc countries.

Table 8: Outputs by **BART** and **BART-Facet** on different facets. Both models are able to generate reasonable summaries given the specified facet. BART-Facet provides more information of Method and less errors than BART (e.g. "IIT" is a typo of "ICT"). However both models tend to directly copy text from the source, for example both outputs of Purpose can be found in the introduction of the paper.

Replicating and Extending "*Because Their Treebanks Leak*": Graph Isomorphism, Covariants, and Parser Performance

Mark Anderson
Universidade da Coruña, CITIC
Department of CS & IT
m.anderson@udc

Anders Søgaard
Dpt. of Computer Science
University of Copenhagen
soegaard@di.ku.dk

Carlos Gómez-Rodríguez
Universidade da Coruña, CITIC
Department of CS & IT
carlos.gomez@udc.es

Abstract

Søgaard (2020) obtained results suggesting the fraction of trees occurring in the test data isomorphic to trees in the training set accounts for a non-trivial variation in parser performance. Similar to other statistical analyses in NLP, the results were based on evaluating linear regressions. However, the study had methodological issues and was undertaken using a small sample size leading to unreliable results. We present a replication study in which we also bin sentences by length and find that only a small subset of sentences vary in performance with respect to graph isomorphism. Further, the correlation observed between parser performance and graph isomorphism in the wild disappears when controlling for covariants. However, in a controlled experiment, where covariants are kept fixed, we do observe a strong correlation. We suggest that conclusions drawn from statistical analyses like this need to be tempered and that controlled experiments can complement them by more readily teasing factors apart.

1 Introduction

We undertake a replication study of Søgaard (2020) which introduced graph isomorphism (DUG - directed unlabelled graph isomorphism) as a means of explaining differences in parser performance across different treebanks. It measures the ratio of graphs[1] in the test set that were also observed in the training data. It is intuitive that this would likely be related to parser performance.

However, DUG has two important covariants. The size of the training data impacts DUG because the smaller a treebank is, the less likely there will be many crossovers between training and test data. DUG is also tied to the mean sentence length in the test data: smaller sentences are much more likely to have a tree structure already seen in the training, as there are fewer possible trees and the reverse is true for longer sentences, e.g. the number of possible trees for a sentence with 20 tokens is 12,826,228.

[1]Note that in the treebanks used in this paper, namely Universal Dependencies, well-formed trees are enforced.

2 Related Work

There is a long history of investigating the causes of variance in parser performance. The effect of training data size on parser performance is well attested (Sagae et al., 2008; Falenska and Çetinoğlu, 2017; Strzyz et al., 2019; Dehouck et al., 2020). Sentence length has also been observed to impact performance (McDonald and Nivre, 2011). One likely factor behind this is different sentence lengths having difference dependency distance distributions (Ferrer-i-Cancho and Liu, 2014) which in turn affects parsing as longer dependencies are typically harder to parse (Anderson and Gómez-Rodríguez, 2020; Falenska et al., 2020). Others have offered explanations based on linguistic characteristics such as morphological complexity (Dehouck and Denis, 2018; Çöltekin, 2020), part-of-speech bigram perplexity (Berdicevskis et al., 2018), and word order freedom (Gulordava and Merlo, 2016).

The history of reproduction and replication in NLP is not so well established, with only a few studies in recent years, e.g. on Universal Dependency (UD) parsing (Çöltekin, 2020) and on automatic essay scoring systems (Huber and Çöltekin, 2020).

Linear techniques, linear regression models or evaluating correlation coefficients are commonly used for statistical analyses of NLP systems. They have been used to model constituency parser performance (Ravi et al., 2008), to evaluate what affects annotation agreement (Bayerl and Paul, 2011), to investigate what impacts statistical MT systems (Guzman and Vogel, 2012), what impacts performance on span identifying tasks (Papay et al., 2020), and many other examples. Therefore, it is likely that lessons drawn from this replication

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1090–1098
August 1–6, 2021. ©2021 Association for Computational Linguistics

	Original			10 seeds		
	CoNLL18	UDPipe 1.2	UDPipe 2.0	CoNLL18	UDPipe 1.2	UDPipe 2.0
Training size	0.014	0.100	0.060	-0.019	-0.346	-0.005
+ DUG	0.228	0.061	0.097	-0.004	-0.553	0.091
+ $\langle L_{\textbf{test}} \rangle$	0.195	0.169	0.146	-0.007	-0.370	0.140
All	-0.078	0.157	0.086	-0.413	-0.138	0.106

Table 1: Issues with using multivariable linear model and cross-validation (CV) to evaluate explained variance. The first set of columns (Original) uses the exact same settings as the original paper (namely one CV split and the original seed) on the original data (CoNLL18) and the predictions from UDPipe 1.2 and UDPipe 2.0 for the extended data. The DUG explained variance is much smaller for the new data. The second set of columns show the same analysis but averaged over 10 different seeds used for the CV splits. The explained variances are almost all negative, which means the linear fit failed.

analysis will be impactful in a broader sense as the conclusions here can be applied in many subareas of NLP, namely the sensitive handling of covariants by using partial coefficients, controlled experiments, or signal subtraction; a strong adherence to visualising data; and considering whether the phenomena under consideration are likely to be sensitive to sentence length, as is often the case in NLP, and if so undertaking a sentence-length binning analysis to complement coarser analyses.

2.1 Original paper

Søgaard (2020) attempted to explain the difference of parser performance across treebanks by using DUG and also undirected unlabelled graph isomorphism (UUG). Two graphs are isomorphic if there is a renaming of vertices that makes them equal. The first process in calculating DUG (or UUG) is to collect the set of unique graphs that occur in the training data. In the original paper, this set of graphs is referred to as the isomorphisms. Once the training isomorphisms are obtained for a given treebank, the number of graphs in the test data that are members of one of these equivalence classes is counted. The final value is then the proportion of test instances that are isomorphic to the training data. This then gives a value between 0 (all test instances are unique) and 1 (no unique test instances).

The analysis was undertaken using a small sample of treebanks that were used at the CoNLL 2018 shared task, using the LAS of the top performing system for each treebank to measure parser performance (Zeman et al., 2018). The impact DUG (or UUG) has on parsing performance was evaluated by fitting a linear regression to the data with DUG as the control variable. A number of other potential measurements that could explain parser performance were also taken into consideration, but only as alternative explanation and not covariants. The exception to this was using the size of the training data as a covariant. The explained variance and absolute error for each linear regression fit was reported using a three-fold cross-validation. The results suggested that DUG was the most strongly correlated measurement evaluated. We show that this result does not hold up when accounting for covariants, that using cross-validation method with the linear regression is not a robust method for an analysis like this, and that by controlling the main covariants of DUG, we can observe a more trustworthy correlation to parser performance.

3 Analysis and results

We evaluate directed graph isomorphism (DUG) as it was more strongly related to parser performance in the original paper.

Main covariants We focus on the two main covariants of DUG: training data size (in sentences) and mean sentence length of the test data, $\langle L_{\text{test}} \rangle$.

Data and parsers The data from the original paper consists of 33 UD treebanks, with LAS taken from the respective top performing parser from the CoNLL 2018 shared task (Zeman et al., 2018). Note that these systems are all variations of the biaffine graph-based parser of Dozat and Manning (2017). For replication, we also use a neural transition-based system UDPipe 1.2 (Straka et al., 2016), using UD models 2.4 and UD v2.5 (Zeman et al., 2019), and a neural graph-based system UDPipe 2.0 (Straka, 2018), using UD models 2.6 and UD v2.7 (Zeman et al., 2020). This results in 94 treebanks for UDPipe 1.2 and 90 for UDPipe 2.0. The difference is due to issues running the web-based UDPipe 2.0 on larger files.

3.1 Reproduction and replication

In the original paper, the analysis focuses on fitting a multi-variable linear regression to the data to control for covariants. However, the models only used training size plus one other variable as features. Further, cross-validation is used so as to avoid over-fitting. While over-fitting isn't directly an issue, the metrics that are typically reported overestimate the variance explained by a linear model, e.g. explained variance, η^2, or R^2 (Lane et al., 2007). Averaging η^2 over different splits can potentially offset this positive bias but it requires a certain amount of data to be reliable. In Table 1, we show the results using the original data from Søgaard (2020). The values shown in the left-most column are exact reproductions of the original values. Only the value for $\langle L_{\text{test}} \rangle$ is different as the original paper appears to have used a normalised value. We also show η^2 for the linear model using all variables, which is negative, i.e. the fit failed.

We next show the results using UDPipe 1.2 and 2.0. While the values for training size on its own and with $\langle L_{\text{test}} \rangle$ are similar, the high η^2 for training size with DUG is no longer observed. This seems to be due to specious results born out of serendipitous splits for the smaller sample from CoNLL 2018.

We then tested this same procedure using different seeds to shuffle the cross-validation splits. The results are almost exclusively negative, i.e. the linear models failed to fit to the data at all. This further highlights an issue of using this methodology when sample size is small, as the random split can have large impact on the statistical metrics.

3.2 Extending the analysis

As the linear models performed so poorly, we measured the correlation coefficients (Spearman's ρ) for each of the variables with respect to LAS and also the potential covariants with respect to DUG. These are reported in Table 2 and we include visualisations of these in Figures 5 and 6 in the Appendix

	CoNLL18	UDPipe 1.2	UDPipe 2.0
size	0.46 (p=0.007)	0.54 (p<0.001)	0.37 (p<0.001)
DUG	-0.13 (p=0.458)	-0.13 (p=0.213)	-0.18 (p=0.083)
$\langle L_{\textbf{test}} \rangle$	0.20 (p=0.272)	0.35 (p=0.001)	0.33 (p=0.001)
size	0.44 (p=0.011)	0.42 (p<0.001)	0.46 (p<0.001)
$\langle L_{\textbf{test}} \rangle$	-0.96 (p<0.001)	-0.91 (p<0.001)	-0.92 (p<0.001)

Table 2: Spearman's ρ for variables with respect to LAS (top) and DUG (bottom).

	CoNLL18	UDPipe 1.2	UDPipe 2.0
log-size	0.055	0.319	0.126
+DUG	0.132	0.410	0.277
+$\langle L_{\textbf{test}} \rangle$	0.106	0.452	0.294
All	-0.184	0.412	0.229

Table 3: Using multivariable linear model and CV to evaluate explained variance with random shuffling (10 splits) and logarithmic transformation of treebank size.

for the CoNLL 2018 data and the UDPipe 2.0 data. Interestingly, DUG has the highest p-value for all systems, far from statistical significance. However, DUG appears to be strongly correlated to both covariants, especially $\langle L_{\text{test}} \rangle$ with $\rho > 0.9$ and $p < 0.001$ for all datasets and systems. Also of note is that training data size is convincingly correlated to LAS, but based on the linear models it doesn't appear to be predictive of parser performance. Based on this and on the visualisation of the data in Figures 5 and 6 in the Appendix (as well as visualisations of training size vs. LAS in the literature, see §2), it seems clear that the relation between these variables is not linear but logarithmic. We show LAS against training data size with a logarithmic scale in Figure 4 in the Appendix.

Table 3 shows the results of the limited linear model and cross-validation technique using 10 different seeds as above and using log training size. For these results, the explained variance of the models are all positive and relatively high, that is, the models manage to fit the data unlike in the original setup. This one change offsets the failure of the linear model technique, which is not surprising. However, it seems to suggest that DUG is not a useful feature, as training size with $\langle L_{\text{test}} \rangle$ outperforms training size with DUG for all datasets except CoNLL18. And the models which use all features are worse than just using training data size and $\langle L_{\text{test}} \rangle$, with the CoNLL18 model resulting in a negative explained variance, again meaning the fit failed. For CoNLL18, training data size and DUG does outperform the model using $\langle L_{\text{test}} \rangle$.

3.3 Sentence length binning

We analyse the relation between test sentence lengths and DUG by binning the data with respect to sentence length. This entails taking each sentence of length l for each treebank, in both the training and test data, and calculating DUG and the corresponding LAS based on these subsets. Figure 1 shows some of these bins (for sentences of

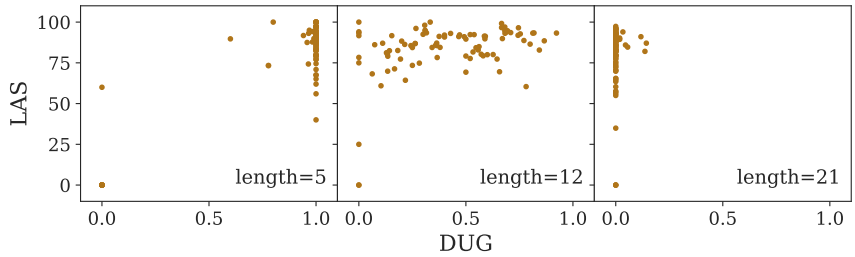


Figure 1: DUG binned wrt sentence length. Values are for UDPipe 2.0 with UD v2.7 for 90 treebanks.

length of 5, 12, and 21 tokens) for UDPipe 2.0. A full visualisation of each bin ranging from length 3 tokens to 30 is shown in Figure 7 in the Appendix.

DUG is almost exclusively 1.0 for shorter sentences, as can be seen in Figure 1 for sentence length 5. The number of possible directed trees for sentences with less tokens is too small for there not to be crossover: there are only 9 possible unlabelled trees for sentences of length 5 (Sloane, 1996). Conversely, for longer sentences, DUG is almost exclusively 0.0 as the number of possible tree structures is considerable (35,221,832 for sentences of length 21).

For a small subset of sentence lengths, ranging from length 9 to 14, there is meaningful spread of values for DUG, with a broadly-speaking linear relation with respect to LAS. Based on this result, i.e. that only certain sentence lengths are suitable for using DUG, we considered using a *focused* version of DUG, i.e. a variant calculated considering only sentences between length 9 and 14 in the training and test data. We then analysed how this measurement correlated with parser performance. Table 4 shows the correlations for focused DUG with respect to LAS, training size, and $\langle L_{\text{test}} \rangle$. While the correlation between focused DUG and LAS is much higher than for DUG and LAS, this is due to the focused version being much more strongly correlated to training size ($\rho = 0.91$ with a p-value less than 0.001 for both datasets) and the correlation with $\langle L_{\text{test}} \rangle$ is much diminished. Also, this focused version of DUG improves performance for the linear model when used only with training data size, but $\langle L_{\text{test}} \rangle$ improves it much more. Using all 3 is again worse than just using training data size with $\langle L_{\text{test}} \rangle$, however, focused DUG doesn't lower the performance as much as the full variant does.

	UDPipe 1.2	UDPipe 2.0
LAS	0.47 (p<0.001)	0.31 (p=0.003)
size	0.91 (p<0.001)	0.91 (p<0.001)
$\langle L_{\textbf{test}} \rangle$	0.32 (p=0.002)	-0.34 (p=0.001)
log-size	0.319	0.126
+DUG	0.331	0.147
+$\langle L_{\textbf{test}} \rangle$	0.452	0.294
All	0.406	0.265

Table 4: Correlations wrt focused DUG (top) and explained variance (bottom) for focused DUG (sentence lengths 9 to 14) with shuffling for CV (10 seeds).

3.4 Controlling covariants

Having established that DUG does not improve linear models predicting LAS and that DUG is strongly correlated to training treebank size and $\langle L_{\text{test}} \rangle$, we attempted to find a signal by removing the background signals associated with these variables. We applied a linear fit to the training data size and LAS and then divided the LAS scores by the predicted values of that fit. Then we applied a linear fit to $\langle L_{\text{test}} \rangle$ and these *normalised* values and again divided these values out. Finally, we evaluated these *doubly normalised* values against DUG. This process is shown in Figure 2 for UDPipe 2.0 and the resulting coefficients for UDPipe 1.2 and 2.0 are in Table 7 of the Appendix. Removing the signals of the covariants results in a linear fit against DUG with a zero gradient and with a coefficient of 0.01 (p=0.926). Removing the variance associated with these covariants effectively removes any signal associated with DUG.

To corroborate this background subtraction analysis, we also report the partial coefficients in Table 5. When controlling for both covariants, correlations are small, and p-values very high, for both

	CoNLL18	UDPipe 1.2	UDPipe 2.0
DUG	-0.13 (p=0.458)	-0.13 (p=0.213)	-0.18 (p=0.083)
size	-0.44 (p=0.010)	-0.50 (p<0.001)	-0.46 (p<0.001)
$\langle L_{\textbf{test}} \rangle$	0.18 (p=0.329)	-0.13 (p=0.213)	0.21 (p=0.049)
both	-0.27 (p=0.126)	0.01 (p=0.915)	-0.12 (p=0.245)

Table 5: Partial Spearman's ρ for DUG with covariants.

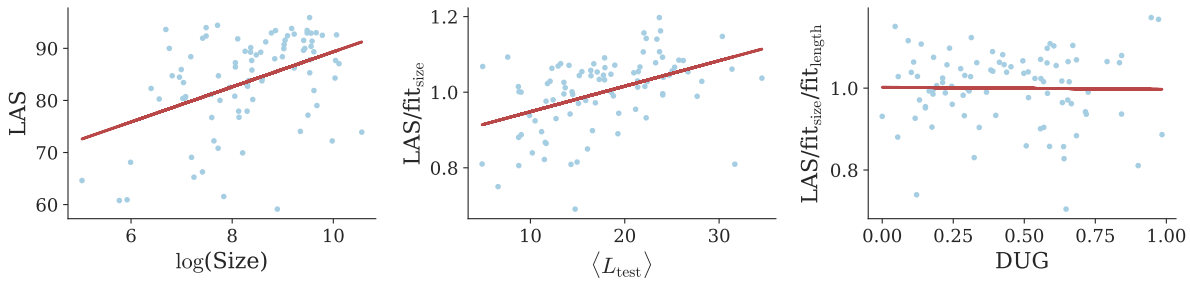


Figure 2: Visualisation of removing background signal associated with covariants of the log of training size (log(Size)) and mean test length $\langle L_{test} \rangle$. The spearman's ρ for DUG and LAS is -0.18 (p=0.083), for DUG and LAS/bcg$_{size}$ is -0.40 (p<0.001) compared to $\langle L_{test} \rangle$and LAS/bcg$_{size}$ of 0.465 (p<0.001), and finally DUG and LAS/bcg$_{size}$bcg$_{Ltest}$ is 0.01 (p=0.926).

UDPipe systems. CoNLL18 has a stronger signal, but it is negative (which is the opposite relation one would expect) and has a large p-value.

3.5 Controlled experiment - fixing covariants

We also evaluated DUG's relation to LAS in a controlled experiment where we sampled subsets of treebanks keeping training data size constant and also the sentence length of both training and test data. We trained UDPipe 1.2 models (UDPipe 2.0 is not available beyond using pre-existing models), using standard settings. We were limited to 9 treebanks, as we required a reasonable amount of data and using only one sentence length reduces the number of usable treebanks. We combined all of the data for treebanks which had over 1200 sentences of length 12. We then created splits such that a single 1000-sentence training set was created by randomly sampling sentences. Then a number of 200-sentence test sets were created, generating as many splits as the data allowed for a given treebank.

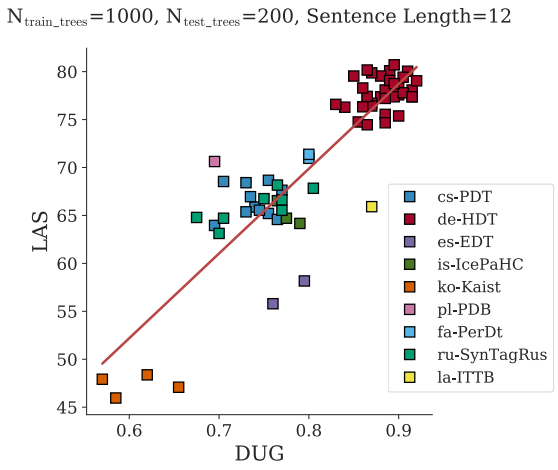


Figure 3: DUG vs LAS for controlled experiment. ρ = 0.82 (p< 0.001).

In this way we varied DUG indirectly, but by using different treebanks to sample from we obtained values spanning a reasonable range (0.6 - 0.9). This results in a Spearman's ρ of 0.82 (p<0.001) and is visualised in Figure 3 in the Appendix. So in this rigid context, we do observe a very strong correlation between DUG and LAS, echoing the analysis from the sentence-length binning procedure.

4 Conclusion

With this case study we have shown the value of replicating analyses in NLP. Our analysis has shown that the original results were unreliable and it has highlighted methodological issues the original analysis had. Also, the results regarding the methodology presented here (i.e. the need to visualise and evaluate correlations before considering linear regression techniques, the potential sensitivity to sentence length of measurements used in NLP statistical analyses, the need to control for all covariants and evaluate their impact using partial coefficients at the very least, and finally that using controlled experiments can help better evaluate the impact of specific measurements and can complement statistical analyses) will likely be useful for other statistical analyses in different areas of NLP.

Acknowledgements

MA and CGR received funding from the European Research Council, under the EU's Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), from MINECO (ANSWER-ASAP, TIN2017-85160-C2-1-R), from Xunta de Galicia (ED431C 2020/11), and from CITIC, funded by Xunta de Galicia and the European Union (ERDF - Galicia 2014-2020 Program), by grant ED431G 2019/01. AS received funding from a Google Focused Research Award.

References

Mark Anderson and Carlos Gómez-Rodríguez. 2020. Inherent dependency displacement bias of transition-based algorithms. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5147–5155, Marseille, France. European Language Resources Association.

Petra Saskia Bayerl and Karsten Ingmar Paul. 2011. What determines inter-coder agreement in manual annotations? A meta-analytic investigation. *Computational Linguistics*, 37(4):699–725.

Aleksandrs Berdicevskis, Çağrı Çöltekin, Katharina Ehret, Kilu von Prince, Daniel Ross, Bill Thompson, Chunxiao Yan, Vera Demberg, Gary Lupyan, Taraka Rama, and Christian Bentz. 2018. Using Universal Dependencies in cross-linguistic complexity research. In *Second Workshop on Universal Dependencies*, pages 8–17, Brussels, Belgium.

Çağrı Çöltekin. 2020. Verification, reproduction and replication of NLP experiments: A case study on parsing Universal Dependencies. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 46–56, Barcelona, Spain (Online). Association for Computational Linguistics.

Mathieu Dehouck, Mark Anderson, and Carlos Gómez-Rodríguez. 2020. Efficient EUD parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 192–205, Online. Association for Computational Linguistics.

Mathieu Dehouck and Pascal Denis. 2018. A framework for understanding the role of morphology in Universal Dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2864–2870, Brussels, Belgium. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. *Proceedings of the 5th International Conference on Learning Representations*.

Agnieszka Falenska, Anders Björkelund, and Jonas Kuhn. 2020. Integrating graph-based and transition-based dependency parsers in the deep contextualized era. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 25–39, Online. Association for Computational Linguistics.

Agnieszka Falenska and Özlem Çetinoğlu. 2017. Lexicalized vs. delexicalized parsing in low-resource scenarios. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 18–24, Pisa, Italy. Association for Computational Linguistics.

Ramon Ferrer-i-Cancho and Haitao Liu. 2014. The risks of mixing dependency lengths from sequences of different length. *Glottotheory*, 5(2):143–155.

Kristina Gulordava and Paola Merlo. 2016. Multilingual dependency parsing evaluation: A large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4:343–356.

Francisco Guzman and Stephan Vogel. 2012. Understanding the performance of statistical MT systems: A linear regression framework. In *Proceedings of COLING 2012*, pages 1029–1044, Mumbai, India. The COLING 2012 Organizing Committee.

Eva Huber and Çağrı Çöltekin. 2020. Reproduction and replication: A case study with automatic essay scoring. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5603–5613, Marseille, France. European Language Resources Association.

David M. Lane, David Scott, Mikki Hebl, Rudy Guerral, Dan Osherson, Heidi Zimmer, et al. 2007. *Online Statistics Education: A Multimedia Course of Study*. Online, Rice University, University of Houston Clear Lake, and Tufts University.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895, Online. Association for Computational Linguistics.

Sujith Ravi, Kevin Knight, and Radu Soricut. 2008. Automatic prediction of parser accuracy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 887–896, Honolulu, Hawaii. Association for Computational Linguistics.

Kenji Sagae, Yusuke Miyao, Rune Saetre, and Jun'ichi Tsujii. 2008. Evaluating the effects of treebank size in a practical application for parsing. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 14–20, Columbus, Ohio. Association for Computational Linguistics.

Neil James Alexander Sloane. 1996. The On-Line Encyclopedia of Integer Sequences. A000081.

Anders Søgaard. 2020. Some languages seem easier to parse because their treebanks leak. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2765–2770, Online. Association for Computational Linguistics.

Milan Straka. 2018. Udpipe 2.0 prototype at CoNLL 2018 UD shared task. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, et al. 2019. Universal Dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Joakim Nivre, et al. 2020. Universal Dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Appendix

The appendix mainly consists of visualisations corresponding to the statistical analyses described in the main body. Some additional information is given to supplement the main analyses in Tables 6 and 7 which give the correlations for the focused DUG analysis and the background removal process, respectively.

Figure 4 shows the logarithmic relation between LAS and the training data size for UDPipe 2.0 and UD v2.7. Figure 5 gives the visualisations for the data used in the original paper and Figure 6 gives the corresponding visualisation for UDPipe 2.0 and UD v2.7.

Figure 7 expands the example plots shown in Figure 1 which only showed extreme cases. This shows LAS versus DUG for every sentence length bin from length 3 to 30. This clearly shows the issue with DUG as discussed in the main body.

All the data used for the analyses presented in this paper can be found in the supplementary material associated with the paper.

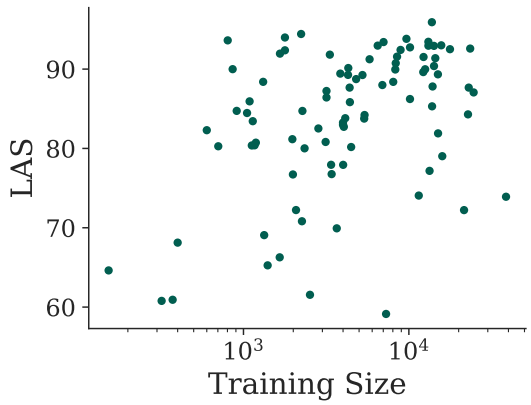


Figure 4: LAS with respect to training set size, in logarithmic scale, for UDPipe 2.0 and UD v2.7.

	UDPipe 1.2	UDPipe 2.0
DUG	0.47 (p<0.001)	0.31 (p=0.003)
size	-0.15 (p=0.153)	-0.10 (p=0.335)
$\langle L_{\textbf{test}} \rangle$	0.64 (p<0.001)	0.48 (p<0.001)
both	0.17 (p=0.110)	0.04 (p=0.683)

Table 6: Partial Spearman's ρ for focused DUG (i.e. using only the measurement for sentences of length 9 to 14) with covariants.

	Spearman's ρ	p-value
DUG LAS	-0.184	0.083
DUG LAS-bcg$_{\textbf{size}}$	-0.400	0.000
DUG LAS-bcg$_{\textbf{size,Ltest}}$	0.010	0.926
$\langle L_{\text{Test}} \rangle$ **LAS-bcg$_{\textbf{size}}$**	0.465	0.000

Table 7: Correlation of DUG with LAS and then with LAS with the background associated with size and length (L) removed. Isolated row shows correlation of LAS without size background and mean sentence length in test data.

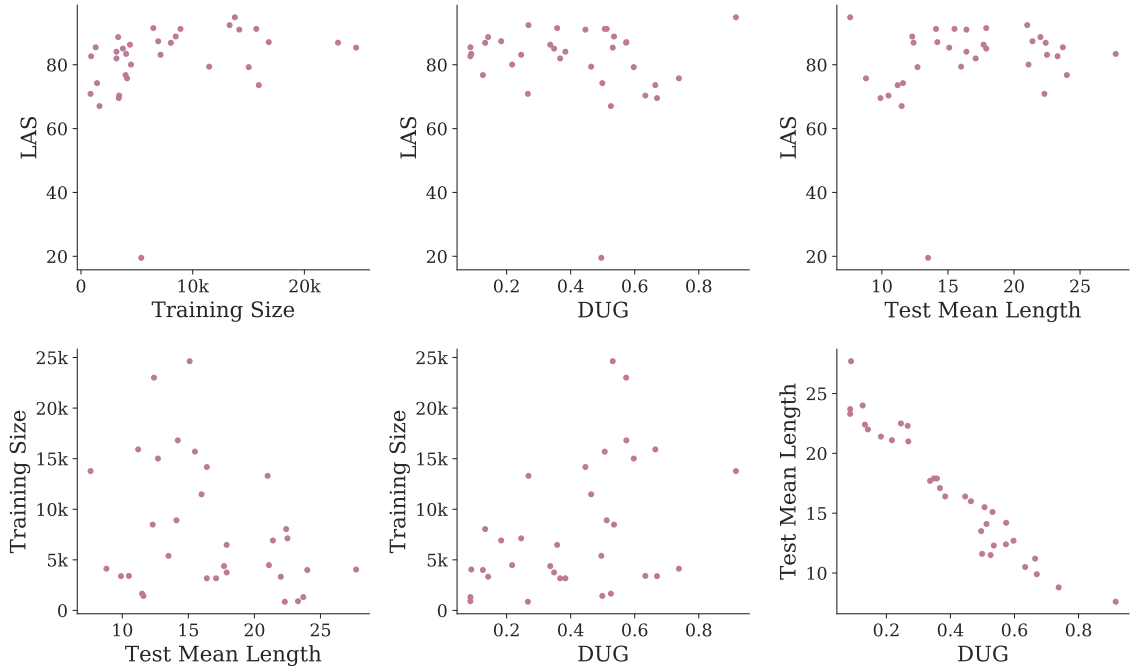


Figure 5: Data from original paper.

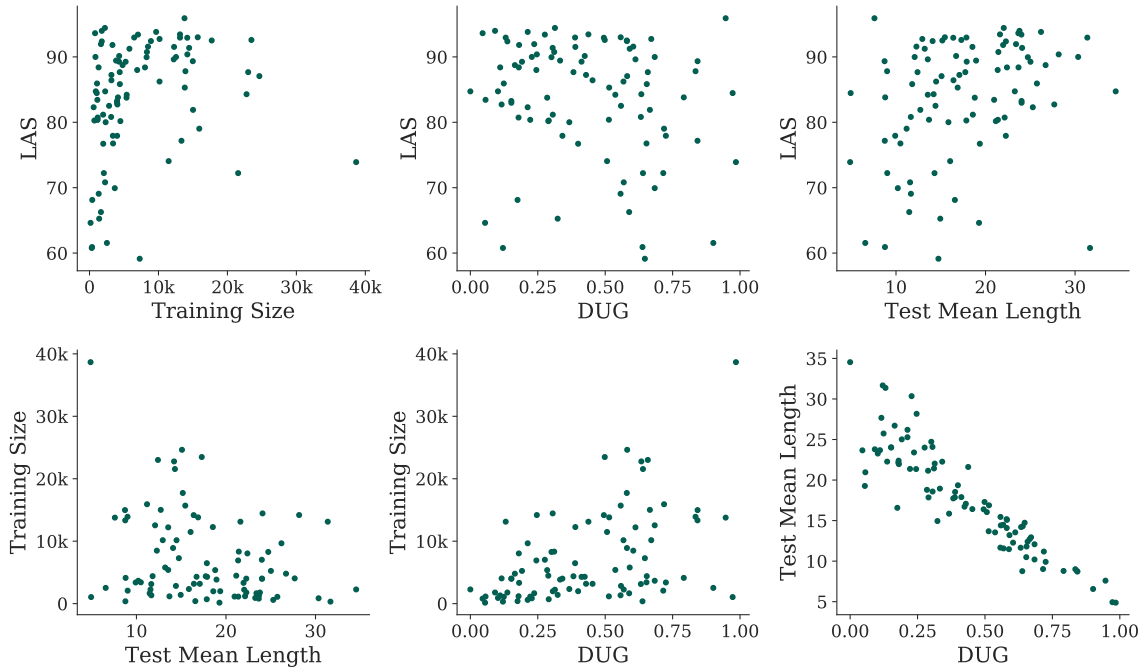


Figure 6: Data for UDPipe 2.0 and UD v2.7 using DUG.

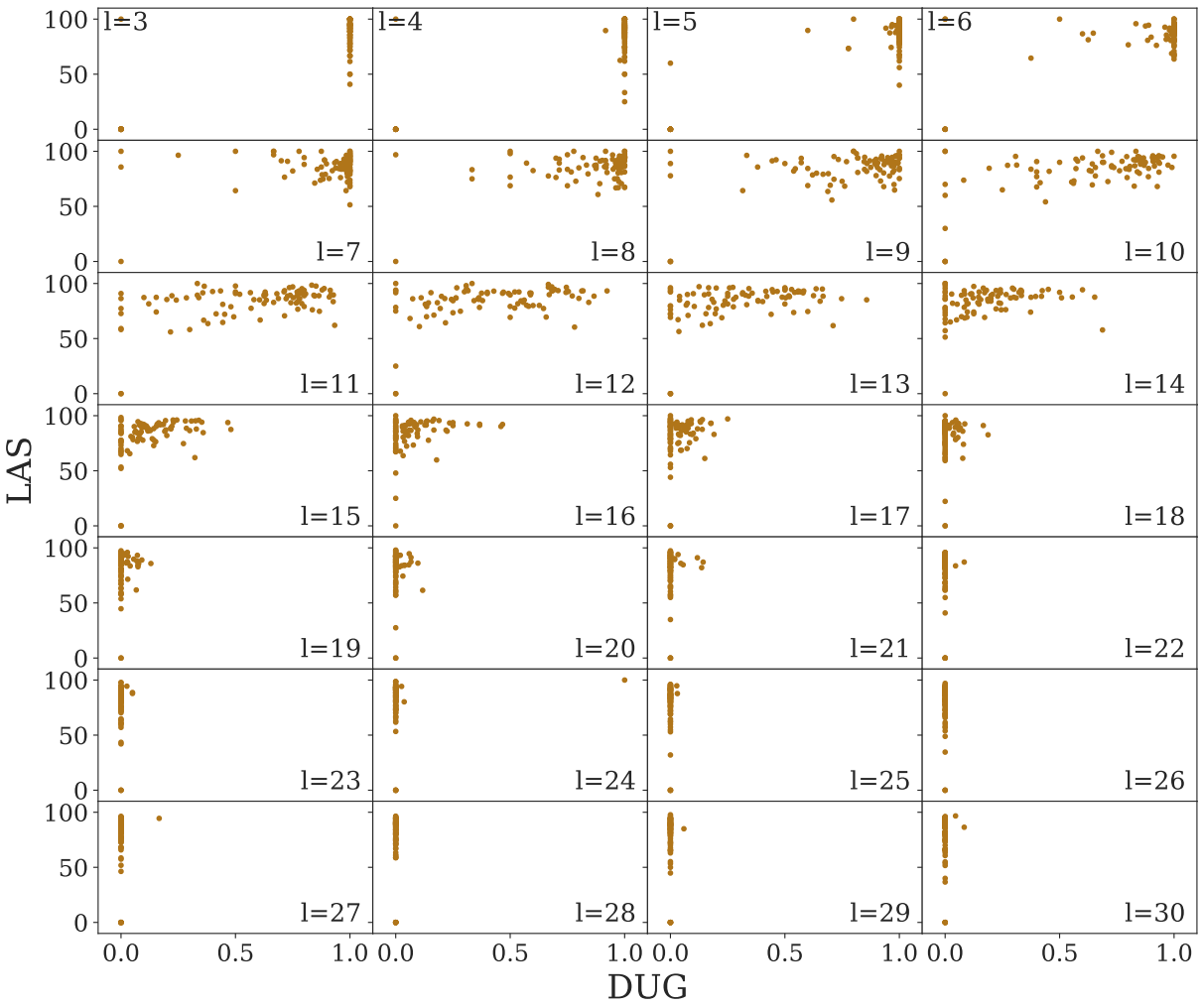


Figure 7: Length-binned analaysis. Data for UDPipe 2.0 and UD v2.7 using DUG.

Don't Rule Out Monolingual Speakers: A Method For Crowdsourcing Machine Translation Data

Rajat Bhatnagar, Ananya Ganesh and **Katharina Kann**
University of Colorado Boulder
{rajat.bhatnagar, ananya.ganesh, katharina.kann}@colorado.edu

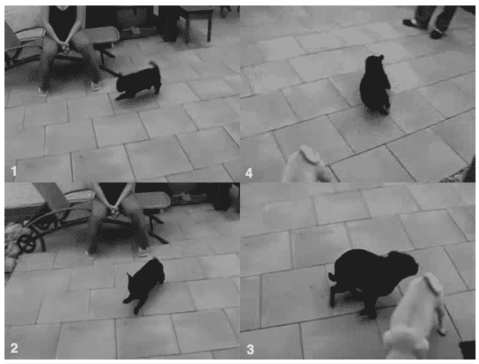
Abstract

High-performing machine translation (MT) systems can help overcome language barriers while making it possible for everyone to communicate and use language technologies in the language of their choice. However, such systems require large amounts of parallel sentences for training, and translators can be difficult to find and expensive. Here, we present a data collection strategy for MT which, in contrast, is cheap and simple, as it does not require bilingual speakers. Based on the insight that humans pay specific attention to movements, we use graphics interchange formats (GIFs) as a pivot to collect parallel sentences from monolingual annotators. We use our strategy to collect data in Hindi, Tamil and English. As a baseline, we also collect data using images as a pivot. We perform an intrinsic evaluation by manually evaluating a subset of the sentence pairs and an extrinsic evaluation by finetuning mBART (Liu et al., 2020) on the collected data. We find that sentences collected via GIFs are indeed of higher quality.

1 Introduction

Machine translation (MT) – automatic translation of text from one natural language into another – provides access to information written in foreign languages and enables communication between speakers of different languages. However, developing high performing MT systems requires large amounts of training data in the form of parallel sentences – a resource which is often difficult and expensive to obtain, especially for languages less frequently studied in natural language processing (NLP), endangered languages, or dialects.

For some languages, it is possible to scrape data from the web (Resnik and Smith, 2003), or to leverage existing translations, e.g., of movie subtitles (Zhang et al., 2014) or religious texts (Resnik et al., 1999). However, such sources of data are only available for a limited number of languages, and it is impossible to collect large MT corpora for a diverse set of languages using these methods. Professional translators, which are a straightforward alternative, are often rare or expensive.



	Sentence for GIFs	Showing only image 2 to annotators
English Annotator	A dog is running around in a circle.	A woman is sitting on a lawn chair while a dog runs nearby.
Hindi Annotator	एक कुत्ता गोल गोल भाग रहा है (A dog is running around in a circle)	काला कुत्ता खेल रहा है। (A black dog is playing)

Figure 1: Sentences written by English and Hindi annotators using GIFs or images as a pivot.

In this paper, we propose a new data collection strategy which is cheap, simple, effective and, importantly, does not require professional translators or even bilingual speakers. It is based on two assumptions: (1) **non-textual modalities can serve as a pivot for the annotation process** (Madaan et al., 2020); and (2) **annotators subconsciously pay increased attention to moving objects**, since humans are extremely good at detecting motion, a crucial skill for survival (Albright and Stoner, 1995). Thus, we propose to leverage graphics interchange formats (GIFs) as a pivot to collect parallel data in two or more languages.

We prefer GIFs over videos as they are short in duration, do not require audio for understanding and describe a comprehensive story visually. Furthermore, we hypothesize that GIFs are better pivots than images – which are suggested by Madaan et al. (2020) for MT data collection – based on our second assumption. We expect that people who

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 1099–1106
August 1–6, 2021. ©2021 Association for Computational Linguistics

are looking at the same GIF tend to focus on the main action and characters within the GIF and, thus, tend to write more similar sentences. This is in contrast to using images as a pivot, where people are more likely to focus on different parts of the image and, hence, to write different sentences, cf. Figure 1.

We experiment with collecting Hindi, Tamil and English sentences via Amazon Mechanical Turk (MTurk), using both GIFs and images as pivots. As an additional baseline, we compare to data collected in previous work (Madaan et al., 2020). We perform both intrinsic and extrinsic evaluations – by manually evaluating the collected sentences and by training MT systems on the collected data, respectively – and find that leveraging GIFs indeed results in parallel sentences of higher quality as compared to our baselines.[1]

2 Related Work

In recent years, especially with the success of transfer learning (Wang et al., 2018) and pretraining in NLP (Devlin et al., 2019), several techniques for improving neural MT for low-resource languages have been proposed (Sennrich et al., 2016; Fadaee et al., 2017; Xia et al., 2019; Lample et al., 2017; Lewis et al., 2019; Liu et al., 2020).

However, supervised methods still outperform their unsupervised and semi-supervised counterparts, which makes collecting training data for MT important. Prior work scrapes data from the web (Lai et al., 2020; Resnik and Smith, 2003), or uses movie subtitles (Zhang et al., 2014), religious texts (Resnik et al., 1999), or multilingual parliament proceedings (Koehn, 2005). However, those and similar resources are only available for a limited set of languages. A large amount of data for a diverse set of low-resource languages cannot be collected using these methods.

For low-resource languages, Hasan et al. (2020) propose a method to convert noisy parallel documents into parallel sentences. Zhang et al. (2020) filter noisy sentence pairs from MT training data.

The closest work to ours is Madaan et al. (2020). The authors collect (pseudo-)parallel sentences with images from the Flickr8k dataset (Hodosh et al., 2013) as a pivot, filtering to obtain images which are simplistic and do not contain culture-specific references. Since Flickr8k already contains 5 English captions per image, they select images whose captions are short and of high similarity to each other. Culture-specific images are manually discarded. We compare to the data from Madaan et al. (2020) in Section 4, denoting it as M20.

3 Experiments

3.1 Pivot Selection

We propose to use GIFs as a pivot to collect parallel sentences in two or more languages. As a baseline, we further collect parallel data via images as similar to our GIFs as possible. In this subsection, we describe our selection of both mediums.

GIFs We take our GIFs from a dataset presented in Li et al. (2016), which consists of 100k GIFs with descriptions. Out of these, 10k GIFs have three English one-sentence descriptions each, which makes them a suitable starting point for our experiments. We compute the word overlap in F1 between each possible combination of the three sentences, take the average per GIF, and choose the highest scoring 2.5k GIFs for our experiments. This criterion filters for GIFs for which all annotators focus on the same main characters and story, and it eliminates GIFs which are overly complex. We thus expect speakers of non-English languages to focus on similar content.

Images Finding images which are comparable to our GIFs is non-trivial. While we could compare our GIFs' descriptions to image captions, we hypothesize that the similarity between the images obtained thereby and the GIFs would be too low for a clean comparison. Thus, we consider two alternatives: (1) using the *first* frame of all GIFs, and (2) using the *middle* frame of all GIFs.

In a preliminary study, we obtain two Hindi one-sentence descriptions from two different annotators for both the first and the middle frame for a subset of 100 GIFs. We then compare the BLEU (Papineni et al., 2002) scores of all sentence pairs. We find that, on average, sentences for the middle frame have a BLEU score of 7.66 as compared to 4.58 for the first frame. Since a higher BLEU score indicates higher similarity and, thus, higher potential suitability as MT training data, we use the middle frames for the image-as-pivot condition in our final experiments.

[1] All data collected for our experiments is available at https://nala-cub.github.io/resources.

Rating	Sentences from the GIF-as-Pivot Setting
1	A child flips on a trampoline. A girl enjoyed while playing.
3	A man in a hat is walking up the stairs holding a bottle of water. A man is walking with a plastic bottle.
5	A man is laughing while holding a gun. A man is laughing while holding a gun.
	Sentences from the Image-as-Pivot Setting
1	A woman makes a gesture in front of a group of other women. This woman is laughing.
3	An older woman with bright lip stick lights a cigarette in her mouth. This woman is lighting a cigarette.
5	A woman wearing leopard print dress and a white jacket is walking forward. A woman is walking with a leopard print dress and white coat.

Table 1: Sentences obtained in English and Hindi for each setting where both annotators agree on the rating. The first sentence is the sentence written in English and the second sentence is the corresponding English translation of the Hindi sentence, translated by the authors.

3.2 Data Collection

We use MTurk for all of our data collection. We collect the following datasets: (1) one single-sentence description in Hindi for each of our 2,500 GIFs; (2) one single-sentence description in Hindi for each of our 2,500 images, i.e., the GIFs' middle frames; (3) one single-sentence description in Tamil for each of the 2,500 GIFs; (4) one single-sentence description in Tamil for each of the 2,500 images; and (5) one single-sentence description in English for each of our 2,500 images. To build parallel data for the GIF-as-pivot condition, we randomly choose one of the available 3 English descriptions for each GIF.

For the collection of Hindi and Tamil sentences, we restrict the workers to be located in India and, for the English sentences, we restrict the workers to be located in the US. We use the instructions from Li et al. (2016) with minor changes for all settings, translating them for Indian workers.[2]

Each MTurk human intelligence task (HIT) consists of annotating five GIFs or images, and we expect each task to take a maximum of 6 minutes. We pay annotators in India \$0.12 per HIT (or \$1.2 per hour), which is above the minimum wage of \$1 per hour in the capital Delhi.[3] Annotators in the US are paid \$1.2 per HIT (or \$12 per hour). We have obtained IRB approval for the experiments reported in this paper (protocol #: 20-0499).

[2]Our instructions can be found in the appendix.

[3]https://paycheck.in/salary/minimumwages/16749-delhi

	GIF-as-Pivot	Image-as-Pivot	M20
Hindi–English	**2.92**	2.20	2.63
Tamil–English	**3.03**	2.33	-

Table 2: Manual evaluation of a subset of our collected sentences; scores from 1 to 5; higher is better.

3.3 Test Set Collection

For the extrinsic evaluation of our data collection strategy we train and test an MT system. For this, we additionally collect in-domain development and test examples for both the GIF-as-pivot and the image-as-pivot setting.

Specifically, we first collect 250 English sentences for 250 images which are the middle frames of previously unused GIFs. We then combine them with the English descriptions of 250 additional unused GIFs from Li et al. (2016). For the resulting set of 500 sentences, we ask Indian MTurk workers to provide a translation into Hindi and Tamil. We manually verify the quality of a randomly chosen subset of these sentences. Workers are paid \$1.2 per hour for this task. We use 100 sentence pairs from each setting as our development set and the remaining 300 for testing.

4 Evaluation

4.1 Intrinsic Evaluation

In order to compare the quality of the parallel sentences obtained under different experimental conditions, we first perform a manual evaluation of a subset of the collected data. For each lan-

	Rating	GIF-as-pivot	Image-as-pivot	M20
Hi-En	5	**13.08**	2.5	10.0
Ta-En		**6.0**	3.5	-
Hi-En	>= 4	**35.77**	15.5	26.43
Ta-En		**37.0**	14.0	-
Hi-En	>= 3	**61.15**	39.0	51.43
Ta-En		**67.5**	42.5	-
Hi-En	>= 2	**82.69**	63.0	75.0
Ta-En		**92.5**	72.5	-
Hi-En	>= 1	**100.0**	**100.0**	**100.0**
Ta-En		**100.0**	**100.0**	-

Table 3: Cumulative percentages with respect to each setting; GIF-as-pivot shows the best results;

guage pair, we select the same random 100 sentence pairs from the GIF-as-pivot and image-as-pivot settings. We further choose 100 random sentence pairs from M20. We randomly shuffle all sentence pairs and ask MTurk workers to evaluate the translation quality. Each sentence pair is evaluated independently by two workers, i.e., we collect two ratings for each pair. Sentence pairs are rated on a scale from 1 to 5, with 1 being the worst and 5 being the best possible score.[4]

Each evaluation HIT consists of 11 sentence pairs. For quality control purposes, each HIT contains one manually selected example with a perfect (for Hindi–English) or almost perfect (for Tamil–English) translation. Annotators who do not give a rating of 5 (for Hindi–English) or a rating of at least 4 (for Tamil–English) do not pass this check. Their tasks are rejected and republished.

Results The average ratings given by the annotators are shown in Table 2. Sentence pairs collected via GIF-as-pivot obtain an average rating of 2.92 and 3.03 for Hindi–English and Tamil–English, respectively. Sentences from the image-as-pivot setting only obtain an average rating of 2.20 and 2.33 for Hindi–English and, respectively, Tamil–English. The rating obtained for M20 (Hindi only) is 2.63. As we can see, for both language pairs the GIF-as-pivot setting is rated consistently higher than the other two settings, thus showing the effectiveness of our data collection strategy. This is in line with our hypothesis that the movement displayed in GIFs is able to guide the sentence writer's attention.

We now explicitly investigate how many of the translations obtained via different strategies are acceptable or good translations; this corresponds to a score of 3 or higher. Table 3 shows that 61.15% of the examples are rated 3 or above in the GIF-as-pivot setting for Hindi as compared to 39.0% and 51.43% for the image-as-pivot setting and M20, respectively. For Tamil, 67.5% of the sentences collected via GIFs are at least acceptable translations. The same is true for only 42.5% of the sentences obtained via images.

We show example sentence pairs with their ratings from the GIF-as-pivot and image-as-pivot settings for Hindi–English in Table 1.

[4]The definitions of each score as given to the annotators can be found in the appendix.

Test Set	Training Set	500	1000	1500	1900	2500
Direction: Hindi to English						
GIF	GIF	**6.41**	**13.06**	**14.39**	**14.81**	**16.09**
GIF	Image	5.71	8.17	9.5	9.7	10.49
GIF	M20	3.19	6.84	7.99	6.9	N/A
Image	GIF	2.93	8.18	9.11	8.84	9.24
Image	Image	**8.46**	**10.05**	**11.15**	**11.25**	**12.14**
Image	M20	1.27	5.79	6.76	6.68	N/A
M20	GIF	1.66	5.21	5.75	6.78	**6.69**
M20	Image	1.63	4.53	4.98	5.09	5.63
M20	M20	**5.08**	**6.96**	**7.23**	**8.23**	N/A
All	GIF	3.47	**8.46**	**9.35**	**9.81**	**10.28**
All	Image	**4.9**	7.28	8.19	8.32	9.04
All	M20	3.37	6.57	7.32	7.37	N/A
Direction: English to Hindi						
GIF	GIF	0.63	1.68	2.01	1.72	**3.07**
GIF	Image	**0.81**	**2.18**	1.43	2.29	1.86
GIF	M20	0.42	2.09	**2.99**	**3.06**	N/A
Image	GIF	0.11	1.19	1.03	0.97	**1.42**
Image	Image	0.15	1.19	1.04	1.09	1.29
Image	M20	**0.22**	**1.23**	**1.95**	**1.68**	N/A
M20	GIF	1.15	2.75	4.25	4.52	4.88
M20	Image	1.32	3.09	4.41	4.1	**5.16**
M20	M20	**5.12**	**12.27**	**12.65**	**13.31**	N/A
All	GIF	0.68	1.96	2.61	2.62	**3.3**
All	Image	0.82	2.25	2.51	2.65	3.01
All	M20	**2.24**	**5.9**	**6.54**	**6.75**	N/A

Table 4: BLEU for different training and test sets; *All* denotes a weighted average over all test sets; all models are obtained by finetuning mBART; best scores for each training set size and test set in bold.

4.2 Extrinsic Evaluation

We further extrinsically evaluate our data by training an MT model on it. Since, for reasons of practicality, we collect only 2,500 examples, we leverage a pretrained model instead of training from scratch. Specifically, we finetune an mBART model (Liu et al., 2020) on increasing amounts of data from all setting in both directions. mBART is

Test Set	Training Set	500	1000	1500	2000	2500
Direction: Tamil to English						
GIF	GIF	**2.63**	**4.46**	**8.26**	**9.27**	**4.99**
GIF	Image	2.33	3.34	3.00	4.77	3.83
Image	GIF	0.95	2.42	3.15	3.67	2.74
Image	Image	**6.65**	**5.62**	**6.02**	**7.75**	**7.22**
All	GIF	1.79	3.44	**5.71**	**6.47**	3.87
All	Image	**4.49**	**4.48**	4.51	6.26	**5.53**
Direction: English to Tamil						
GIF	GIF	0	**0.54**	**1.00**	**0.83**	**0.84**
GIF	Image	**0.5**	0.18	0.96	0.43	0.48
Image	GIF	0	0.31	0.36	**0.62**	**0.7**
Image	Image	**0.41**	**0.35**	**0.51**	0.36	0.29
All	GIF	0	**0.43**	0.68	**0.73**	**0.77**
All	Image	**0.46**	0.27	**0.74**	0.4	0.39

Table 5: BLEU for different training and test sets; *All* denotes a weighted average over all test sets; all models are obtained by finetuning mBART; best scores for each training set size and test set in bold.

a transformer-based sequence-to-sequence model which is pretrained on 25 monolingual raw text corpora. We finetune it with a learning rate of 3e-5 and a dropout of 0.3 for up to 100 epochs with a patience of 15.

Results The BLEU scores for all settings are shown in Tables 4 and 5 for Hindi–English and Tamil–English, respectively. We observe that increasing the dataset size mostly increases the performance for all data collection settings, which indicates that the obtained data is useful for training. Further, we observe that each model performs best on its own in-domain test set.

Looking at Hindi-to-English translation, we see that, on average, models trained on sentences collected via GIFs outperform sentences from images or M20 for all training set sizes, except for the 500-examples setting, where image-as-pivot is best. However, results are mixed for Tamil-to-English translation.

Considering English-to-Hindi translation, models trained on M20 data outperform models trained on sentences collected via GIFs or our images in nearly all settings. However, since the BLEU scores are low, we manually inspect the obtained outputs. We find that the translations into Hindi are poor and differences in BLEU scores are often due to shared individual words, even though the overall meaning of the translation is incorrect. Similarly, for English-to-Tamil translation, all BLEU scores are below or equal to 1. We thus conclude that 2,500 examples are not enough to train an MT system for these directions, and, while we report all results here for completeness, we believe that the intrinsic evaluation paints a more complete picture.[5] We leave a scaling of our extrinsic evaluation to future work.

5 Conclusion

In this work, we made two assumptions: (1) that a non-textual modality can serve as a pivot for MT data collection, and (2) that humans tend to focus on moving objects. Based on this, we proposed to collect parallel sentences for MT using GIFs as pivots, eliminating the need for bilingual speakers and reducing annotation costs. We collected parallel sentences in English, Hindi and Tamil using our approach and conducted intrinsic and extrinsic evaluations of the obtained data, comparing our strategy to two baseline approaches which used images as pivots. According to the intrinsic evaluation, our approach resulted in parallel sentences of higher quality than either baseline.

Acknowledgments

We would like to thank the anonymous reviewers, whose feedback helped us improve this paper. We are also grateful to Aman Madaan, the first author of the M20 paper, for providing the data splits and insights from his work. Finally, we thank the members of CU Boulder's NALA Group for their feedback on this research.

References

Thomas D Albright and Gene R Stoner. 1995. Visual motion perception. *Proceedings of the National Academy of Sciences*, 92(7):2433–2440.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural

[5] We also manually inspect the translations into English: in contrast to the Hindi translations, most sentences at least partially convey the same meaning as the reference.

machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Guokun Lai, Zihang Dai, and Yiming Yang. 2020. Unsupervised parallel corpus mining on web data. *arXiv preprint arXiv:2009.08595*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A new dataset and benchmark on animated GIF description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Aman Madaan, Shruti Rijhwani, Antonios Anastasopoulos, Yiming Yang, and Graham Neubig. 2020. Practical comparable data collection for low-resource languages via images. *arXiv preprint arXiv:2004.11954*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

P. Resnik, M. Olsen, and Mona T. Diab. 1999. The bible as a parallel corpus: Annotating the 'book of 2000 tongues'. *Computers and the Humanities*, 33:129–153.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8545–8554, Online. Association for Computational Linguistics.

Shikun Zhang, Wang Ling, and Chris Dyer. 2014. Dual subtitles as parallel corpora. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1869–1874, Reykjavik, Iceland. European Language Resources Association (ELRA).

A Sentence Rating Instructions

Score	Title	Description
1	Not a translation	There is no relation whatsoever between the source and the target sentence
2	Bad	Some word overlap, but the meaning isn't the same
3	Acceptable	The translation conveys the meaning to some degree but is a bad translation
4	Good	The translation is missing a few words but conveys most of the meaning adequately
5	Perfect	The translation is perfect or close to perfect

Table 6: Description of the ratings for the manual evaluation of translations.

B MTurk Instructions

Instructions for English image task

Below you will see five images. Your task is to describe each image in one English sentence. You should focus solely on the visual content presented in the image. Each sentence should be grammatically correct. It should describe the main characters and their actions, but NOT your opinions, guesses or interpretations.

- DOs
  - Please use only English words. No digits allowed (spell them out, e.g., three).
  - Sentences should neither be too short nor too long. Try to be concise.
  - Each sentence must contain a verb.
  - If possible, include adjectives that describe colors, size, emotions, or quantity.
  - Please pay attention to grammar and spelling.
  - Each sentence must express a complete idea, and make sense by itself.
  - The sentence should describe the main characters, actions, setting, and relationship between the objects.
- DONTs
  - The sentence should NOT contain any digits.
  - The sentence should NOT mention the name of a movie, film, and character.
  - The sentence should NOT mention invisible objects and actions.
  - The sentence should NOT make subjective judgments about the image.

Remember, please describe only the visual content presented in the images. Focus on the main characters and their actions.

निर्देश **(Instructions for GIF Task in Hindi)**

नीचे आपको पांच गिफ (GIF) दिखाई देंगे। आपको हर गिफ को एक वाक्य में हिंदी में समझाना है। आपको सिर्फ गिफ में जो हो रहा है उसपर ध्यान देना है। आपके वाक्य की व्याकरण सही होनी चाहिए। आपको मुख्य पात्रों और उनके कार्यों का वर्णन करना है और आपको अपनी राय नहीं देनी है।

- क्या करें -
  - कृपया केवल हिंदी शब्दों और हिंदी लिपि (देवनागरी) का उपयोग करें। किसी भी अंक को पूरी तरह लिखे (उदहारण - तीन लिखे नाकि ३)।
  - वाक्य न तो बहुत छोटे होने चाहिए और न ही बहुत लंबे। संक्षिप्त होने का प्रयास करें। वाक्य कम से कम चार शब्दों का होना चाहिए।
  - प्रत्येक वाक्य में एक क्रिया होनी चाहिए।
  - यदि संभव हो तो विशेषणों का इस्तिमाल करें जो की रंगो, आकार व भावनाओं को अच्छे से समझा सके।
  - कृपया व्याकरण और स्पेलिंग पर ध्यान दें।
  - प्रत्येक वाक्य को एक पूर्ण विचार व्यक्त करना चाहिए, और खुद से समझ में आना चाहिए।
  - आपके वाक्य को मुख्य अतिथिओ, वस्तुओ और उनके साथ हो रही चीज़ो को समझाना है।
- क्या न करें -
  - वाक्य में कोई अंक नहीं होना चाहिए।
  - वाक्य में किसी फिल्म या एक्टर का नाम नहीं होना चाहिए।
  - वाक्य में अदृश्य वस्तुओं और कार्यों का उल्लेख नहीं होना चाहिए।
  - वाक्य में अपने व्यक्तिगत राय न डालें।

याद रखें, गिफ में जो दिख रहा है उसी के बारे में लिखे। मुख्य पात्रों और उनके कार्यों पर ध्यान दें।

வழிமுறைகள் **(Instructions for GIF task in Tamil)**

கீழே ஐந்து அனிமேஷன் செய்யப்பட்ட கிப் (GIF) காட்டப்பட்டுள்ளன. ஒவ்வொரு GIF ஐ ஒரு தமிழ் வாக்கியத்தில் விவரிப்பதே உங்கள் பணி. GIF இல் உள்ள காட்சியில் மட்டுமே நீங்கள் கவனம் செலுத்த வேண்டும். GIF இல் உள்ள முக்கிய கதாபாத்திரங்களையும் அவற்றின் செயல்களையும் வர்ணிக்க வேண்டும், ஆனால் உங்கள் கருத்துக்கள், யூகங்கள் அல்லது விளக்கங்கள் அல்ல. ஒவ்வொரு வாக்கியமும் இலக்கணப்படி சரியாக இருக்க வேண்டும்.

- செய்க:
  - தமிழ் வார்தைகளை மட்டும் பயன்படுத்தவும். எண்களை வார்த்தையில் எழுதவும் (3 -> மூன்று)
  - வாக்கியங்கள் மிகக் குறுகியதாகவோ அல்லது நீண்டதாகவோ இருக்கக்கூடாது.
  - ஒவ்வொரு வாக்கியத்திலும் ஒரு வினை (செயலை குறிக்கும் வார்த்தை) இருக்க வேண்டும்.
  - முடிந்தால், வண்ணங்கள், அளவு, உணர்ச்சிகளை விவரிக்கும் வார்த்தைகளை சேர்க்கவும்.
  - இலக்கணம் மற்றும் எழுத்துப்பிழைகள் இல்லாதப்படி எழுதவும்.
  - ஒவ்வொரு வாக்கியமும் முழுமையாக இருக்க வேண்டும், மேலும் வாக்கியத்தை தனியாகப் படித்தால் அர்த்தம் புரிய வேண்டும்.।
  - வாக்கியம் GIFஇல் உள்ள முக்கிய கதாபாத்திரங்கள், செயல்கள், அமைப்பு, மற்றும் பொருள்களுக்கு இடையிலான உறவை விவரிக்க வேண்டும்.
- செய்யாதீர்:
  - வாக்கியத்தில் எந்த எண்களும் இருக்கக்கூடாது.
  - வாக்கியத்தில் எந்த திரைப்படம், மற்றும் நடிகர் அல்லது கதாபாத்திரத்தின் பெயரைக் குறிப்பிடக்கூடாது.
  - வாக்கியத்தில் கண்ணுக்கு தெரியாத பொருள்கள் மற்றும் செயல்களைக் குறிப்பிடக்கூடாது.
  - வாக்கியத்தில் தங்களின் எண்ணங்களோ தீர்மானங்களோ இருக்கக்கூடாது.

கவனிக்கவும்: அனிமேஷன் செய்யப்பட்ட GIF இல் காணும் காட்சியை மட்டும் வர்ணிக்கவும். அதில் இருக்கும் முக்கிய கதாப்பாத்திரங்கள் மற்றும் செயல்களில் கவனம் செலுத்தவும்.

Figure 2: Instructions for the data collection via images in English, via GIFs in Hindi and Tamil.

Author Index