4th Workshop on Computational Approaches to Discourse

Proceedings of the Workshop

July 13-14, 2023
Preface

Welcome to the 4th Workshop on Computational Approaches to Discourse, CODI!

CODI provides a venue to bring together researchers working on all aspects of discourse in Computational Linguistics and NLP. Our aim is to provide a venue for the entire discourse processing community where we can present and exchange our theories, algorithms, software, datasets, and tools.

The workshop consists of invited talks, contributed papers, extended abstracts, and ACL Findings presentations. We received paper submissions that span a wide range of topics, addressing issues related to discourse representation and parsing, reference and coreference resolution, summarization, dialogue, pragmatics, applications, and more. As the workshop is hybrid this year, papers are presented live either in person or remotely and discussed during live Q&A sessions.

We are pleased that CODI 2023 features the third edition of the DISRPT (DIScourse Relation Parsing and Treebanking) shared task on Discourse Segmentation, Connective and Relation Identification across Formalisms. As we hope that the next CODI workshops will also feature shared tasks and other special events, the workshop also includes a discussion on future shared tasks, special sessions on discourse representation and parsing, coreference resolution, and multilingual discourse processing and machine translation.

We thank our invited speakers, Yufang Hou, IBM Research Ireland and adjunct senior lecturer and co-supervisor at UKP Lab-TU Darmstadt, who works on referential discourse modeling, argument mining, and scholarly document processing; and Giuseppe Carenini, University of British Columbia, known for his work on discourse parsing, summarization, and generation. We would also like to thank our reviewers for their thoughtful and instructive comments. They helped us to prepare an excellent and inclusive workshop program. Finally we would like to thank the ACL 2023 workshop chairs Eduardo Blanco, Yang Feng, and Annie Louis who organized the ACL workshops program.

The CODI Organizers,

Chloé Braud, Christian Hardmeier, Junyi Jessy Li, Sharid Loáiciga, Michael Strube, and Amir Zeldes
Program Committee

Chairs

Chloé Braud, IRIT - CNRS - ANITI
Christian Hardmeier, IT University of Copenhagen
Junyi Jessy Li, University of Texas at Austin
Sharid Loáiciga, University of Gothenburg
Michael Strube, Heidelberg Institute for Theoretical Studies
Amir Zeldes, Georgetown University

Program Committee

Katherine Atwell, University of Pittsburgh
Giuseppe Carenini, university of british columbia
Haixia Chai, Heidelberg Institute for Theoretical Studies gGmbH
Jackie Chi Kit Cheung, Mila / McGill University
Vera Demberg, Saarland University
Pascal Denis, INRIA
Elisa Ferracane, Abridge AI, Inc.
Zhengxian Gong, Computer Science and Technology School, Soochow University
Jie He, University of Edinburgh
Ryuichiro Higashinaka, Nagoya University/NTT
SungHo Jeon, Heidelberg Institute for Theoretical Studies
Prathyusha Jwalapuram, Rakuten
Murathan Kurfalı, Stockholm University
Ekaterina Lapshinova-Koltunski, Stiftung Universität Hildesheim
Ramesh Manuvinakurike, Intel labs
Philippe Muller, IRIT, University of Toulouse
Maciej Ogrodniczuk, Institute of Computer Science, Polish Academy of Sciences
Carolyn Rosé, Carnegie Mellon University
Manfred Stede, University of Potsdam
Francielle Vargas, University of São Paulo
Bonnie Webber, University of Edinburgh
Deniz Zeyrek, Middle East Technical University
Heike Zinsmeister, Universität Hamburg
# Table of Contents

**MuLMS-AZ: An Argumentative Zoning Dataset for the Materials Science Domain**  
Timo Schrader, Teresa Bürkle, Sophie Henning, Sherry Tan, Matteo Finco, Stefan Grünewald, Maira Indrikova, Felix Hildebrand and Annemarie Friedrich .................................................. 1

**A Side-by-side Comparison of Transformers for Implicit Discourse Relation Classification**  
Bruce W. Lee, Bongseok Yang and Jason Lee ................................................................. 16

**Ensemble Transfer Learning for Multilingual Coreference Resolution**  
Tuan Lai and Heng Ji ......................................................................................... 24

**Contrastive Hierarchical Discourse Graph for Scientific Document Summarization**  
Haopeng Zhang, Xiao Liu and Jiawei Zhang ............................................................. 37

**Leveraging Structural Discourse Information for Event Coreference Resolution in Dutch**  
Loïc De Langhe, Orphee De Clercq and Veronique Hoste ...................................... 48

**Entity Coreference and Co-occurrence Aware Argument Mining from Biomedical Literature**  
Boyang Liu, Viktor Schlegel, Riza Batista-navarro and Sophia Ananiadou ............. 54

**A Weakly-Supervised Learning Approach to the Identification of Alternative Lexicalizations in Shallow Discourse Parsing**  
René Knaebel ........................................................................................................ 61

**Entity-based SpanCopy for Abstractive Summarization to Improve the Factual Consistency**  
Wen Xiao and Giuseppe Carenini ........................................................................... 70

**Discourse Information for Document-Level Temporal Dependency Parsing**  
Jingcheng Niu, Victoria Ng, Erin Rees, Simon De Montigny and Gerald Penn ........ 82

**Encoding Discourse Structure: Comparison of RST and QUD**  
Sara Shahmohammadi, Hannah Seemann, Manfred Stede and Tatjana Scheffler .......... 89

**Exploiting Knowledge about Discourse Relations for Implicit Discourse Relation Classification**  
Nobel Varghese, Frances Yung, Kaveri Anuranjana and Vera Demberg .................. 99

**SAE-NTM: Sentence-Aware Encoder for Neural Topic Modeling**  
Hao Liu, Jingsheng Gao, Suncheng Xiang, Ting Liu and Yuzhuo Fu ...................... 106

**Improving Long Context Document-Level Machine Translation**  
Christian Herold and Hermann Ney ....................................................................... 112

**Unpacking Ambiguous Structure: A Dataset for Ambiguous Implicit Discourse Relations for English and Egyptian Arabic**  
Ahmed Ruby, Sara Stymne and Christian Hardmeier ..................................................... 126

**Two-step Text Summarization for Long-form Biographical Narrative Genre**  
Avi Bleiweiss .......................................................................................................... 145

**The distribution of discourse relations within and across turns in spontaneous conversation**  
S. Magali López Cortez and Cassandra L. Jacobs ......................................................... 156

**Embedding Mental Health Discourse for Community Recommendation**  
Hy Dang, Bang Nguyen, Noah Ziems and Meng Jiang ............................................ 163

**APA-RST: A Text Simplification Corpus with RST Annotations**  
Freya Hewett ......................................................................................................... 173
Program

Thursday, July 13, 2023

09:15 - 09:30  Opening Remarks

09:30 - 10:30  Invited Talk - Yufang Hou: Bridging Resolution: A Journey Towards Modelling Referential Discourse Entities

10:30 - 11:00  Coffee Break

11:00 - 12:20  Session - Summarization/Generation

  Contrastive Hierarchical Discourse Graph for Scientific Document Summarization
  Haopeng Zhang, Xiao Liu and Jiawei Zhang

  Entity-based SpanCopy for Abstractive Summarization to Improve the Factual Consistency
  Wen Xiao and Giuseppe Carenini

  Two-step Text Summarization for Long-form Biographical Narrative Genre
  Avi Bleiweiss

  Improving Long Context Document-Level Machine Translation
  Christian Herold and Hermann Ney

12:30 - 14:00  Lunch

14:00 - 14:50  Session - Coreference

  Ensemble Transfer Learning for Multilingual Coreference Resolution
  Tuan Lai and Heng Ji

  Leveraging Structural Discourse Information for Event Coreference Resolution in Dutch
  Loic De Langhe, Orphee De Clercq and Veronique Hoste

  Entity Coreference and Co-occurrence Aware Argument Mining from Biomedical Literature
  Boyang Liu, Viktor Schlegel, Riza Batista-navarro and Sophia Ananiadou
Thursday, July 13, 2023 (continued)

14:50 - 15:30  Session - Discourse Relations/Other

Unpacking Ambiguous Structure: A Dataset for Ambiguous Implicit Discourse Relations for English and Egyptian Arabic
Ahmed Ruby, Sara Stymne and Christian Hardmeier

Embedding Mental Health Discourse for Community Recommendation
Hy Dang, Bang Nguyen, Noah Ziems and Meng Jiang

15:30 - 16:00  Coffee Break

16:00 - 16:35  Session - Rhetorical Structure Theory

Encoding Discourse Structure: Comparison of RST and QUD
Sara Shahmohammadi, Hannah Seemann, Manfred Stede and Tatjana Scheffler

APA-RST: A Text Simplification Corpus with RST Annotations
Freya Hewett

16:35 - 17:25  Session - Other

Discourse Information for Document-Level Temporal Dependency Parsing
Jingcheng Niu, Victoria Ng, Erin Rees, Simon De Montigny and Gerald Penn

SAE-NTM: Sentence-Aware Encoder for Neural Topic Modeling
Hao Liu, Jingsheng Gao, Suncheng Xiang, Ting Liu and Yuzhuo Fu
Friday, July 14, 2023

09:30 - 10:30  Invited Talk - Giuseppe Carenini: Discourse Processing in the era of Large Language Models

10:30 - 11:00  Coffee Break

11:00 - 12:10  Discourse Relation Parsing and Treebanking (DISRPT) Shared Task

12:10 - 12:30  Discussion of Future Shared Tasks

12:30 - 14:00  Lunch

14:00 - 15:20  Session - Discourse Relations/Other

  MuLMS-AZ: An Argumentative Zoning Dataset for the Materials Science Domain
  Timo Schrader, Teresa Bürkle, Sophie Henning, Sherry Tan, Matteo Finco, Stefan Grünewald, Maira Indrikova, Felix Hildebrand and Annemarie Friedrich

  A Side-by-side Comparison of Transformers for Implicit Discourse Relation Classification
  Bruce W. Lee, Bongseok Yang and Jason Lee

  A Weakly-Supervised Learning Approach to the Identification of Alternative Lexicalizations in Shallow Discourse Parsing
  René Knaebel

  Exploiting Knowledge about Discourse Relations for Implicit Discourse Relation Classification
  Nobel Varghese, Frances Yung, Kaveri Anuranjana and Vera Demberg

  The distribution of discourse relations within and across turns in spontaneous conversation
  S. Magalí López Cortez and Cassandra L. Jacobs

15:20 - 17:00  Coffee and Poster Session

  Chinese-DiMLex: A Lexicon of Chinese Discourse Connectives
  Shujun Wan, Peter Bourgonje, Hongling Xiao, Clara Wan Ching Ho and Manfred Stede
Discourse Structure Extraction from Pre-Trained and Fine-Tuned Language Models in Dialogues
Chuyuan Li, Patrick Huber, Wen Xiao, Maxime Amblard, Chloé Braud and Giuseppe Carenini

Discourse Analysis via Questions and Answers: Parsing Dependency Structures of Questions Under Discussion
Wei-jen Ko, Yating Wu, Cutter Dalton, Dananjay Srinivas, Greg Durrett and Junyi Jessy Li

SWiPE: A Dataset for Document-Level Simplification of Wikipedia Pages
Philippe Laban, Jesse Vig, Wojciech Kryscinski, Shafiq Joty, Caiming Xiong and Chien-sheng Wu

Replicate and Compare with Humans: LLMs Represent Partial Semantic Knowledge in Pronoun Interpretation
Suet-ying Lam, Qingcheng Zeng, Kexun Zhang, Chenyu You and Rob Voigt

Context-Aware Document Simplification
Liam Cripwell, Jol Legrand and Claire Gardent

An Email Dataset for Analyzing Large-Group Decision-Making
Mladen Karan, Prashant Khare, Ravi Shekhar, Stephen Mcquistin, Colin Perkins, Ignacio Castro, Gareth Tyson, Patrick Healey and Matthew Purver

GUMSum: Multi-Genre Data and Evaluation for English Abstractive Summarization
Yang Janet Liu and Amir Zeldes

An Integrated Approach for Political Bias Prediction and Explanation Based on Discursive Structure
Nicolas Devatine, Philippe Muller and Chloé Braud

2 * n is better than n^2: Decomposing Event Coreference Resolution into Two Tractable Problems
Shafiuddin Rehan Ahmed, Abhijnan Nath, James H. Martin and Nikhil Krishnaswamy

Imagination is All You Need! Curved Contrastive Learning for Abstract Sequence Modeling Utilized on Long Short-Term Dialogue Planning
Justus-jonas Erker, Stefan Schaffer and Gerasimos Spanakis

17:00 - 17:30 Closing Remarks and Awards
MuLMS-AZ: An Argumentative Zoning Dataset for the Materials Science Domain

Timo Pierre Schrader1,6 Teresa Bürkle2 Sophie Henning1,3 Sherry Tan4 Matteo Finco2 Stefan Grünewald1,5 Maira Indrikova2 Felix Hildebrand2 Annemarie Friedrich6

1Bosch Center for Artificial Intelligence, Renningen, Germany
2Robert Bosch GmbH, Stuttgart, Germany 3LMU Munich, Germany 4TU Darmstadt, Germany 5University of Stuttgart, Germany 6University of Augsburg, Germany
timo.schrader|teresa.buerkle|sophie.henning@de.bosch.com annemarie.friedrich@informatik.uni-augsburg.de

Abstract

Scientific publications follow conventionalized rhetorical structures. Classifying the Argumentative Zone (AZ), e.g., identifying whether a sentence states a Motivation, a Result or Background information, has been proposed to improve processing of scholarly documents. In this work, we adapt and extend this idea to the domain of materials science research. We present and release a new dataset of 50 manually annotated research articles. The dataset spans seven sub-topics and is annotated with a materials-science focused multi-label annotation scheme for AZ. We detail corpus statistics and demonstrate high inter-annotator agreement. Our computational experiments show that using domain-specific pre-trained transformer-based text encoders is key to high classification performance. We also find that AZ categories from existing datasets in other domains are transferable to varying degrees.

1 Introduction

In academic writing, it is custom to adhere to a rhetorical argumentation structure to convince readers of the relevance of the work to the field (Swales, 1990). For example, authors typically first indicate a gap in prior work before stating the goal of their own research. Argumentative Zoning (AZ) is a natural language processing (NLP) task in which sentences are classified according to their argumentative roles with varying granularity (Teufel et al., 1999, 2009). AZ information can then be used for summarization (Teufel and Moens, 2002; El-Ebshihiy et al., 2020), improved citation indexing (Teufel, 2006), or writing assistance (Feltrim et al., 2006).

Manually annotated AZ datasets (Teufel et al., 1999; Fisas et al., 2016; Soldatova and Liakata, 2007) only exist for few domains and employ differing annotation schemes. The resulting models are not directly applicable to our domain of interest, materials science research, which presents a challenging domain for current NLP methods (e.g., Mysore et al., 2019; Friedrich et al., 2020; O’Gorman et al., 2021). In this paper, we present MuLMS-AZ, the first dataset annotated for AZ in this domain. Working together with domain experts, we derive a hierarchical multi-label annotation scheme (see Table 1). Our scheme includes domain-specific labels such as descriptions of the materials’ Preparation and Characterization, which are crucial distinctions also for NLP applications from the domain experts’ view.

This resource paper makes the following contributions:

- We present a dataset of 50 scientific articles (more than 10,000 sentences) in the domain of materials science manually annotated by domain experts with a hierarchical fine-grained annotation scheme for AZ with high agreement. The corpus will be publicly released.\(^1\)
- We apply several neural models to our dataset that will serve as strong baselines for future work using our new benchmark. We find (a) that using domain-specific pre-trained transformers is key to a successful model, (b) that multi-task learning with existing AZ datasets leads to small benefits, and (c) that the effectiveness of transfer learning of materials science AZ labels from other corpora differs by label.

\(^1\)https://github.com/boschresearch/mulms-az-codi2023

<table>
<thead>
<tr>
<th>Label</th>
<th>Count</th>
<th>Label</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTIVATION</td>
<td>363</td>
<td>EXPLANATION</td>
<td>603</td>
</tr>
<tr>
<td>BACKGROUND</td>
<td>3155</td>
<td>RESULTS</td>
<td>2953</td>
</tr>
<tr>
<td>- PRIORWORK</td>
<td>1824</td>
<td>CONCLUSION</td>
<td>680</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>2579</td>
<td>HEADING</td>
<td>702</td>
</tr>
<tr>
<td>- PREP.</td>
<td>962</td>
<td>CAPTION</td>
<td>485</td>
</tr>
<tr>
<td>- CHARACT.</td>
<td>1347</td>
<td>METADATA</td>
<td>210</td>
</tr>
</tbody>
</table>

Table 1: MuLMS-AZ label counts (multi-label).

Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023), pages 1–15
July 13-14, 2023 ©2023 Association for Computational Linguistics
2 Related Work

In this section, we describe related work on AZ.

AZ Datasets. Table 2 shows the statistics of several related datasets. Three larger-scale datasets manually annotated with AZ information are the AZ-CL corpus (Teufel et al., 1999; Teufel and Moens, 1999), consisting of computational linguistics papers, the Dr. Inventor Multi-Layer Scientific Corpus (DRI, Fisas et al., 2016, 2015), featuring computer graphics papers, and, closest to our domain, the ART corpus (Soldatova and Liakata, 2007), covering topics in physical chemistry and biochemistry. Appendix E explains these datasets in more detail. Teufel et al. (2009) also apply and adapt the annotation scheme of the AZ-CL corpus to the chemistry domain. Accuusto et al. (2021) label sentences with argumentation-related categories (e.g., proposal, means, observation). Making use of sentence-wise author-provided keywords, a dataset of about 388k abstracts with silver standard rhetorical role annotations has been derived from PubMed/MEDLINE (de Moura and Feltrim, 2018).

Modeling. AZ has been modeled as a sentence classification task using maximum entropy models (Teufel and Kan, 2009), SVMs, and CRFs (Guo et al., 2011) leveraging a variety of word, grammatical, heuristic, and discourse features (Guo et al., 2013). Ensemble-based classifiers have also been shown to be effective (Badie et al., 2018; Asadi et al., 2019). LSTM-based models relying on word embeddings have been applied to AZ and to the fundamentally very similar task of assigning zones to sentences in job ads (Liu, 2017; de Moura and Feltrim, 2018; Gnehm and Clematide, 2020). BERT-style (Devlin et al., 2019) models work well for AZ (Mo et al., 2020; Brack et al., 2022). Multi-task training has been found to be beneficial for these models both in-domain (Lauscher et al., 2018) as well as cross-domain (Brack et al., 2021).

Datasets in the Materials Science Domain. Several datasets targeting the domain of materials science research have recently been released. Mysore et al. (2019) annotate paragraphs describing synthesis procedures with graph structures capturing relations and typed arguments. Friedrich et al. (2020) mark similar graph structures corresponding to experiment information in 45 open-access publications. Several works and datasets address named entity recognition in the domain (Yamaguchi et al., 2020; O’Gorman et al., 2021).

<table>
<thead>
<tr>
<th>Dataset</th>
<th># docs</th>
<th># sents</th>
<th># labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ-CL</td>
<td>80</td>
<td>12818</td>
<td>7</td>
</tr>
<tr>
<td>ART</td>
<td>225</td>
<td>34995</td>
<td>11</td>
</tr>
<tr>
<td>DRI</td>
<td>40</td>
<td>10784</td>
<td>10</td>
</tr>
<tr>
<td>MulS-AZ</td>
<td>50</td>
<td>10186</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 2: Manually annotated AZ corpora.

3 Data Sources and Annotated Corpus

In this section, we present our new dataset.

Source of Texts and Preprocessing. We select 50 scientific articles licensed under CC-BY from seven sub-areas of materials science: electrolys, graphene, polymer electrolyte fuel cell (PEMFC), solid oxide fuel cell (SOFC), polymers, semiconductors, and steel. The four SOFC papers were selected from the SOFC-Exp corpus (Friedrich et al., 2020). 11 papers were selected from the OA-STM corpus and classified into the above subject areas by a domain expert. The majority of the papers were found via PubMed and DOAJ using queries prepared by a domain expert. For the OA-STM data, we use the sentence segmentation provided with the corpus, which has been created using GENIA tools (Tsuruoka and Tsujii, 2005). For the remaining texts, we rely on the sentence segmentation provided by INCEpTION v21.0 (Klie et al., 2018) with some manual fixes.

Annotation Scheme. AZs are functional sentence types, i.e., they capture the rhetorical function of a sentence. Together with several domain experts, we design a hierarchical scheme tailored to the materials science domain as shown in Table 3. In addition, we provide ABSTRACT, HEADING, METADATA, CAPTION, FIGURE/TABLE annotations for structural information. We assume a multi-label setting in which annotators may assign any number of labels to a sentence. Our detailed guidelines are available with our dataset.

Corpus Statistics. Documents are rather long (on average 203.7 sentences per document with a standard deviation of ±73.2). There is a tendency towards long sentences (28.7 tokens per sentence on average), but with high variation of ±17.9 due to, e.g., short headings. Table 1 shows how often each AZ label occurs. When ignoring tags for structural information 8133 sentences have exactly one AZ label (or the AZ label and its supertype), 1056 sentences have two labels, and 11 sentences have 3
labels. Labels are similarly distributed across data splits (see Appendix D).

**Inter-Annotator Agreement.** Our entire dataset has been annotated by a single annotator, a graduate student of materials science, who was also involved in the design of the annotation scheme. We compare the annotations of this main annotator to those of another annotator who holds a Master’s degree in materials science and a PhD in engineering. The agreement study is performed on 5 documents (357 sentences). Due to the multi-label scenario, following Krippendorff (1980) we measure $\kappa$ (Cohen, 1960) for each label separately, comparing whether each annotator used a particular label on an instance or not (see Table 4). Our annotators achieve “substantial” agreement (Landis and Koch, 1977) on most labels, “perfect” agreement on identifying HEADINGS (see also Appendix D). Lower, though still “moderate”, agreement on MOTIVATION, EXPLANATION and CONCLUSION can in part be explained by their lower frequency which makes it generally harder to obtain high $\kappa$-values. Intuitively, they also have a more difficult nature compared to the other tags, e.g., we observe disagreements regarding what constitutes a MOTIVATION or an EXPLANATION versus what is purely reporting BACKGROUND. The full confusion matrix and a discussion of agreement on subtags are given in Appendix D; a discussion of multi-label examples can be found in Appendix F.

Our scores are in the same ballpark as those reported by Teufel et al. (1999) on a similar annotation task. For their 7-way task, they report $\kappa$ scores around 0.71-0.75, with differences between categories in one-vs-all measurements ranging from about 0.49 to 0.78. In sum, we conclude that agreement on AZ is satisfactory in our dataset.

<table>
<thead>
<tr>
<th>AZ Label</th>
<th>$\kappa$</th>
<th>AZ Label</th>
<th>$\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>META DATA</td>
<td>0.76</td>
<td>BACKGROUND</td>
<td>0.75</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>0.78</td>
<td>RESULTS</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Table 4: IAA for AZ on 357 sentences.

**4 Modeling**

We model AZ as a multi-label classification problem, using BERT (Devlin et al., 2019) as the underlying text encoder. We also test domain-specific pre-trained variants of BERT. SciBERT (Beltagy et al., 2019) has been pre-trained on articles in the scientific domain. MatSciBERT (Gupta et al., 2022) is a version of SciBERT further pre-trained on materials science articles. We use the CLS embedding as input to a linear layer, transform logits using a sigmoid function and choose labels if their respective score exceeds 0.5. For multi-task experiments with other datasets, we use a single shared language model and one linear output layer per dataset. For hyperparameters, see Appendix A.

As shown in Table 1, the dataset suffers from strong class imbalance. Classifiers tend to under-perform on minority labels (Johnson and Khoshgoftaar, 2019). To address this problem, we apply the multi-label random oversampling (ML-ROS, Charte et al., 2015) algorithm during training. The main idea behind ML-ROS is to dynamically duplicate instances of minority classes while taking the multi-label nature of the problem into account. In a nutshell, the algorithm performs several oversampling iterations, keeping track of the imbalance ratios associated with each label and choosing instances that carry minority labels until a predefined number of additional samples have been chosen. Details are given in Appendix B.
<table>
<thead>
<tr>
<th>Method</th>
<th>LM</th>
<th>mic.-F1</th>
<th>mac.-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Oversampling</td>
<td>BERT</td>
<td>72.6 ±1.0</td>
<td>65.5 ±0.7</td>
</tr>
<tr>
<td></td>
<td>MatSciBERT</td>
<td>76.3 ±0.7</td>
<td>70.1 ±0.7</td>
</tr>
<tr>
<td></td>
<td>SciBERT</td>
<td>76.2 ±0.9</td>
<td>70.2 ±0.6</td>
</tr>
<tr>
<td>ML-ROS</td>
<td>SciBERT</td>
<td>76.7 ±0.7</td>
<td>70.5 ±0.9</td>
</tr>
<tr>
<td></td>
<td>+ MultiTask ART</td>
<td>75.0 ±0.9</td>
<td>68.9 ±1.1</td>
</tr>
<tr>
<td></td>
<td>+ MultiTask AZ-CL</td>
<td>77.2 ±0.3</td>
<td>71.1 ±0.5</td>
</tr>
<tr>
<td>human agreement*</td>
<td></td>
<td>78.7</td>
<td>74.9</td>
</tr>
</tbody>
</table>

Table 5: AZ classification results on MuLMS-AZ test set. *Not directly comparable: computed on documents from agreement study (see Appendix D).

5 Experimental Results

We here detail our experimental results.

Settings. We split our corpus into train, dev, and test sets of 36, 7, and 7 documents. For all experiments and for hyperparameter tuning, we always train five models. The training data is split into five folds. Similar to cross-validation, we train on four folds and use the fifth fold for model selection (cf. van der Goot, 2021), repeating this process five times (also for hyperparameter tuning). The dev set is only used for tuning, and we report scores for the five models on test. In this setting, deviations are naturally higher than when reporting results for the same training data. For hyperparameters and implementation details, see Appendix A. To evaluate our experiments, we use hierarchical precision, recall, and F1 (Silla and Freitas, 2011). These scores operate on the sets of labels per instance, always including the respective supertypes of gold or predicted labels.

Results. Table 5 shows the performance of our neural models on MuLMS-AZ. Overall, the categories can be learned well, approaching our estimate of human agreement. SciBERT clearly outperforms BERT, i.e., using domain-specific embeddings is a clear advantage. However, MatSciBERT does not add upon SciBERT. We hence conduct the remaining experiments using SciBERT. Using ML-ROS results in minor increases for most labels (see also Appendix G). When multi-task learning with the AZ-CL dataset (using 40% of its samples), further increases are observed. It is worth noting that multi-task training with ART does not result in increases although the chemistry domain should be much closer to our domain. This might indicate that despite the domain discrepancy, AZ annotations in AZ-CL are more compatible with ours.

As a first step to explaining what part of rhetorical information can be induced based on data from other corpora, we perform a transfer learning experiment. We carefully manually map the AZ labels of the various datasets (see Appendix E) to the coarse-grained categories used by PubMed (PM). Using these mapped labels, we train binary classifiers that aim to detect the presence of a particular PM label. As training data, we use ART, DRI, and a selection of documents from the PM dataset by de Moura and Feltrim (2018) that were published in materials science journals (see Appendix C). We add AZ-CL to the training data only if an unambiguous mapping of its categories to the PM label in question is possible. Here, we use the dev set of MuLMS-AZ for model selection and hyperparameter tuning. Results for running the resulting classifiers on MuLMS-AZ are reported in Table 6. For BACKGROUND and RESULTS, we observe a high precision, which indicates that similar rhetorical elements may be used. OBJECTIVE and METHOD seem to differ most across datasets as they are likely very domain- and problem-specific. When training with mapped labels on the entire MuLMS-AZ, we observe much higher recall scores across all label groups, again indicating the usefulness of our in-domain training data.

6 Conclusion and Outlook

We have presented a new AZ corpus in the field of materials science annotated by domain experts with high agreement. Our experimental results demonstrate that strong classifiers can be learned on the data and that AZ labels can be transferred from related datasets only to a limited extent.

Our new dataset opens up new research opportunities on cross-domain AZ, class imbalance scenarios, and integrating AZ information in information extraction tasks in materials science.
Limitations

This resource paper describes the dataset in detail, providing strong baselines and first initial cross-domain experiments. It does not aim to provide an extensive set of experiments on cross-domain argumentative zoning yet.

The entire dataset is only singly-annotated. The agreement study was performed on complete documents and hence has only limited data for several labels. Due to the limited funding of the project, we could double-annotate the entire dataset.

Finally, we only test one model class (BERT-based transformers). A potential next step is to test a bigger variety of models and embeddings. Because AZ labels are interdependent within a document, especially document-level models or CRF-based models are promising methods to try. We have also tested only one method (multi-label random oversampling) to deal with the strong class imbalance in the dataset. We have not yet tested further such methods (Henning et al., 2023) or data augmentation methods.

Ethical Considerations

We took care of potential license issue of the data underlying our dataset by exclusively selecting open-access articles published under CC BY.

The main annotator was paid above the minimum wage of our country in the context of a full-time internship. The annotator was aware of the goal of the study and consents to the public release of the data. The remaining domain experts participated on a voluntary basis due to their interest in the topic.

References


Larisa Soldatova and Maria Liakata. 2007. An ontology methodology and cisp—the proposed core information about scientific papers. *JISC Project Report*.


Appendix

A Hyperparameters

We implement all our models using PyTorch. We use AdamW (Loshchilov and Hutter, 2019) as the optimizer for all our models and set the batch size to 16/32 depending on what works best and GPU restrictions. The learning rate stays constant after a linear warmup phase. We set a dropout rate to 0.1 for the linear layer that takes the contextualized embeddings that are produced by BERT as input. Early stopping is applied if the micro-F1 score has not improved for more than 3 epochs. Binary cross entropy is the loss function for the MuLMS-AZ output layer, whereas cross entropy is the loss function used for optimizing the multi-task output heads corresponding to the other AZ datasets. Table 7 lists the various learning rates found during grid search.

<table>
<thead>
<tr>
<th>Method</th>
<th>LM</th>
<th>Learning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Oversampling</td>
<td>BERT</td>
<td>3e-6</td>
</tr>
<tr>
<td></td>
<td>MatSciBERT</td>
<td>8e-6</td>
</tr>
<tr>
<td></td>
<td>SciBERT</td>
<td>3e-6</td>
</tr>
<tr>
<td>ML-ROS</td>
<td>SciBERT</td>
<td>2e-6</td>
</tr>
<tr>
<td>+ MT (+PM)</td>
<td>SciBERT</td>
<td>7e-6</td>
</tr>
<tr>
<td>+ MT (+ART)</td>
<td>SciBERT</td>
<td>2e-6</td>
</tr>
<tr>
<td>+ MT (+AZ-CL)</td>
<td>SciBERT</td>
<td>2e-6</td>
</tr>
<tr>
<td>+ MT (+DRI)</td>
<td>SciBERT</td>
<td>1e-6</td>
</tr>
<tr>
<td>+ MT (+ART+AZ+DRI)</td>
<td>SciBERT</td>
<td>8e-6</td>
</tr>
<tr>
<td>Data Augm. (+PM)</td>
<td>SciBERT</td>
<td>8e-6</td>
</tr>
</tbody>
</table>

Table 7: Learning rates of the different model reported in Table 12

B Multi-Label Random Oversampling (ML-ROS) Algorithm

Figure 1 details our adaption of the multi-label random oversampling (ML-ROS) algorithm originally proposed by Charte et al. (2015). In the initialization (lines 3-7), for each label, all the instances that carry a particular label are collected in what Charte et al. call a bag. The main part of the algorithm (lines 10-24) does the following: For each label $y$, the Imbalance Ratio per label (IRLbl), which is the ratio between the count of the most frequent label and the count of $y$, is calculated:

$$\text{IRLbl}(y) = \frac{\max_{y' \in L} \sum_{i=1}^{|D|} h(y', Y_i)}{\sum_{i=1}^{|D|} h(y, Y_i)}$$

$D$ is the dataset, $L$ is the label set, $Y_i$ is the set of labels assigned to the $i$-th sample and $h$ is an indicator function evaluating if $y \in Y_i$. Hence, the larger the value, the less frequently $y$ occurs compared to the most frequent label.

The per-label values are then used to determine the mean imbalance ratio (MeanIR):

$$\text{MeanIR} = \frac{1}{|L|} \sum_{y' \in L} \text{IRLbl}(y')$$

For each of the labels with an imbalance ratio exceeding the current MeanIR, a random instance of this label is duplicated.

The main part is repeated until the pre-specified size of the oversampled dataset is reached. Our implementation differs from Charte et al. in that we update meanIR in each iteration step and also oversample labels originally not being a minority label when their IRLbl exceeds MeanIR at the beginning of an iteration step.

C List of Materials Science Journals

We used the list of materials-science related journals collected on Wikipedia to filter for abstracts in the PubMed Medline corpus published in journals.\(^5\)

D Further Corpus Statistics for MuLMS-AZ

Table 8 gives the counts of sentences carrying a particular AZ label. Distributions are similar across data splits. Table 8 also lists counts for ABSTRACT, which we decide to exclude from our modeling experiments because including it resulted in performance decreases due to confusion with other labels. Locating the abstract in a document can usually be solved in rule-based ways as abstracts of publications are commonly available in a machine-readable format.

During annotation, we introduced two subtypes of EXPLANATION, HYPOTHESIS and ASSUMP-OTION, distinguishing between scientific hypotheses and assumptions made by the author in cases where often choices are possible. As the overall counts

Inputs: <Dataset> D, <Percentage> P
Outputs: Oversampled dataset
samplesToDuplicate ← ⌊D⌋/100 + P ⋅ P % size increment
L ← labelsInDataset(D) # Obtain the full set of labels
for each label in L do # Bags of samples for each label
    BagLabel ← getSamplesPerLabel(label)
end for
while samplesToDuplicate > 0 do # Loop duplicating instances
    MeanIR ← calculateMeanIR(D, L)
    # Gather minority bags (bag: all instances of a given label)
    minBags = []
    for each label in L do
        IRLabel ← calculateIRPerLabel(D, label)
        if IRLabel > MeanIR then
            minBags += BagLabel
        end if
    end for
    # Duplicate a random sample from each minority bag
    for each minBag in minBags do
        x ← random(1, |minBag|)
        duplicateSample(minBag, x)
    end for
end while

Figure 1: Pseudocode for adapted (dynamic) ML-ROS algorithm.

and agreement were low, we decided to only use the supertype EXPLANATION in all experiments.

Figure 2a shows the label coincidence matrix between the two annotators in the inter-annotator agreement study, i.e., how often each pair of labels co-occurred on an instance. For all labels except MOTIVATION, the majority of coincidences occur on the diagonal. RESULTS is the label most mixed up with others, possibly because these sentences often are long and also contain interpretative information of the other rhetorical types.

Figure 2a breaks this information down the level including subtypes. CHARACTERIZATION and PREPARATION are rarely confused by the domain experts. Similarly, BACKGROUND and PRIOR-WORK are reliably distinguished.

### Agreement on sub-labels.

Our agreement study contained only 12 CAPTION instances. Data inspection showed that the additional (not the main) annotator neglected to use this tag where appropriate, using only content-related tags on these instances. There were also not enough instances of the subtypes PREPARATION and EXPERIMENT CHARACTERIZATION to measure agreement. On identifying the subtype BACKGROUND_PRIORWORK, annotators achieve a κ of 0.8, with (minor) disagreements mainly with regard to using BACKGROUND or its subtype.

<table>
<thead>
<tr>
<th>Label</th>
<th>total</th>
<th>train</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTIVATION</td>
<td>363</td>
<td>273</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>BACKGROUND</td>
<td>3155</td>
<td>2423</td>
<td>440</td>
<td>292</td>
</tr>
<tr>
<td>-PRIORWORK</td>
<td>1824</td>
<td>1387</td>
<td>265</td>
<td>172</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>2579</td>
<td>1896</td>
<td>394</td>
<td>289</td>
</tr>
<tr>
<td>-CHARACTERIZATION</td>
<td>1347</td>
<td>982</td>
<td>200</td>
<td>165</td>
</tr>
<tr>
<td>-PREPARATION</td>
<td>962</td>
<td>705</td>
<td>146</td>
<td>111</td>
</tr>
<tr>
<td>EXPLANATION</td>
<td>603</td>
<td>430</td>
<td>91</td>
<td>82</td>
</tr>
<tr>
<td>RESULTS</td>
<td>2953</td>
<td>2146</td>
<td>440</td>
<td>367</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>680</td>
<td>507</td>
<td>106</td>
<td>67</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>269</td>
<td>190</td>
<td>28</td>
<td>51</td>
</tr>
<tr>
<td>CAPTION</td>
<td>485</td>
<td>309</td>
<td>91</td>
<td>85</td>
</tr>
<tr>
<td>HEADING</td>
<td>702</td>
<td>536</td>
<td>96</td>
<td>70</td>
</tr>
<tr>
<td>METADATA</td>
<td>210</td>
<td>142</td>
<td>40</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 8: Label counts on the complete dataset and on data split subsets. Multi-label counts: Number of sentences in which the label is present. Due to multi-labeling, the sum of these columns exceeds the total amount of sentences. For hierarchical labels, the super-label count includes all sub-label counts.

### Agreement on HEADING.

As it should be straightforward to identify headings, we looked at the 6 cases that one annotator labeled as HEADING but not the other. We found 4 cases to result from broken formatting. One METADATA sentence was wrongly labeled HEADING, and the remaining HEADING sentence was missed by the other annotator.
E Description and Comparison of AZ Datasets.

In this section, we provide a detailed description and comparison of existing AZ datasets. The various corpora try to capture very similar information. However, each corpus defines its set of labels in a slightly different way. Table 10 lists the various labels used in the datasets and groups labels used for the same or very similar purpose. Table 11 shows the label distributions of the corpora.

AZ-CL corpus. The Argumentative Zoning (AZ, Teufel et al., 1999; Teufel and Moens, 1999) corpus\(^7\) consists of 80 manually annotated open-access computational linguistics research articles. Sentences are marked according to their argumentative zone or rhetorical function as one of the following classes: AIM, BACKGROUND, BASIS, CONTRAST, OTHER, OWN or TEXT. Inter-annotator agreement is reported as substantial ($\kappa = 0.71$). The distribution of classes is quite skewed towards OTHER and OWN.

ART corpus. The ART corpus\(^8\) (Soldatova and Liakata, 2007) covers topics in physical chemistry and biochemistry. Articles are annotated according to the CISP/CoreSC annotation scheme (Liakata and Soldatova, 2008). Sentences are

---

\(^7\)https://github.com/WING-NUS/RAZ

\(^8\)https://www.aber.ac.uk/en/cs/research/cb/projects/art/art-corpus/
Table 10: AZ Corpus Zones Mapping and Descriptions. Compare to Table 3.

<table>
<thead>
<tr>
<th>PubMed</th>
<th>AZ-CL</th>
<th>ART</th>
<th>DRI</th>
<th>MulMS-AZ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTIVE</td>
<td>AIM</td>
<td>HYPOTHESIS MOTIVATION GOAL</td>
<td>CHALLENGE MOTIVATION</td>
<td>A sentence describing the research target, goal, aim or the motivation for the research.</td>
<td></td>
</tr>
<tr>
<td>BACKGROUND BACKGROUNDBACKGROUND BACKGROUND BACKGROUNDPRIOR WORK</td>
<td>BACKGROUNDBACKGROUNDCONTRAST BASIS</td>
<td>BACKGROUND EXPERIMENT PREPARATION CHARACTERIZATION EXPLANATION</td>
<td>BACKGROUND EXPERIMENT</td>
<td>A sentence describing the research procedure, models used, or observations made during the research.</td>
<td></td>
</tr>
<tr>
<td>METHOD OWN</td>
<td>OWN</td>
<td>OBJECT, METHOD MODEL EXPERIMENT OBSERVATION</td>
<td>APPROACH EXPERIMENT</td>
<td>A sentence describing the study findings, effects, consequences, and/or analysis of the results.</td>
<td></td>
</tr>
<tr>
<td>RESULT OWN</td>
<td>OWN</td>
<td>RESULT OUTCOME</td>
<td>OUTCOME</td>
<td>A statement concerning the knowledge domain or previous related work.</td>
<td></td>
</tr>
<tr>
<td>CONCLUSION OWN</td>
<td>OWN</td>
<td>CONCLUSION OUTCOME FUTUREWORK</td>
<td>CONCLUSION</td>
<td>A statement concerning the support or rejection of the hypothesis or suggestions of future research.</td>
<td></td>
</tr>
<tr>
<td>–</td>
<td>TEXT OTHER</td>
<td>SENTENCE UNSPECIFIED</td>
<td></td>
<td>Example sentences, broken sentences, etc.</td>
<td></td>
</tr>
</tbody>
</table>

labeled with one of the categories HYPOTHESIS, MOTIVATION, GOAL OF INVESTIGATION, BACKGROUND, OBJECT OF INVESTIGATION, RESEARCH METHOD, MODEL, EXPERIMENT, OBSERVATION, RESULT or CONCLUSION. The annotation scheme also defines subcategories for some of these. The corpus has been annotated by domain experts. In a preliminary study, $\kappa$ was measured as 0.55; however, for the final corpus, only the annotators that had the highest average agreement were selected. Hence, the agreement in the final corpus is expected to be higher.

**DRI corpus.** The Dr. Inventor Multi-Layer Scientific Corpus (DRI, Fisas et al., 2016, 2015), contains 40 scientific articles taken from the domain of computer graphics. Each of the 10,784 sentences was annotated with one of the rhetorical categories: CHALLENGE, BACKGROUND, APPROACH, OUTCOME, or FUTUREWORK. They have also included two other categories SENTENCE for sentences that are characterized by segmentation or character encoding errors and UNSPECIFIED for sentences where identification is not possible. Also to note was the possibility to annotate a combination of two different categories as seen in the example of: OUTCOME_CONTRIBUTION, CHALLENGE_GOAL and CHALLENGE_HYPOTHESIS. The dataset contains close to 5 million sentences. The dataset is not particularly challenging: a simple CRF model achieves an F-score of 93.75, an LSTM-based model achieves 94.77 according to de Moura and Feltrim (2018).

**F Examples**

In this section, we present and discuss several examples from our dataset.

**F.1 Example Sentences**

- **Motivation:** Therefore, it is highly desirable to develop an innovative technology to raise the mass activity of Ir-based OER catalysts to the targeted level.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Label</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ-CL</td>
<td>OWN</td>
<td>8624</td>
</tr>
<tr>
<td></td>
<td>OTHER</td>
<td>2019</td>
</tr>
<tr>
<td></td>
<td>BACKGROUND</td>
<td>789</td>
</tr>
<tr>
<td></td>
<td>CONTRAST</td>
<td>600</td>
</tr>
<tr>
<td></td>
<td>AIM</td>
<td>313</td>
</tr>
<tr>
<td></td>
<td>BASIS</td>
<td>246</td>
</tr>
<tr>
<td></td>
<td>TEXT</td>
<td>227</td>
</tr>
<tr>
<td>ART</td>
<td>RESULT</td>
<td>7373</td>
</tr>
<tr>
<td></td>
<td>BACKGROUND</td>
<td>6657</td>
</tr>
<tr>
<td></td>
<td>OBSERVATION</td>
<td>4659</td>
</tr>
<tr>
<td></td>
<td>METHOD</td>
<td>3751</td>
</tr>
<tr>
<td></td>
<td>MODEL</td>
<td>3456</td>
</tr>
<tr>
<td></td>
<td>CONCLUSION</td>
<td>3083</td>
</tr>
<tr>
<td></td>
<td>EXPERIMENT</td>
<td>2841</td>
</tr>
<tr>
<td></td>
<td>OBJECT</td>
<td>1190</td>
</tr>
<tr>
<td></td>
<td>HYPOTHESIS</td>
<td>656</td>
</tr>
<tr>
<td></td>
<td>GOAL</td>
<td>548</td>
</tr>
<tr>
<td></td>
<td>MOTIVATION</td>
<td>466</td>
</tr>
<tr>
<td>DRI</td>
<td>APPROACH</td>
<td>5038</td>
</tr>
<tr>
<td></td>
<td>BACKGROUND</td>
<td>1760</td>
</tr>
<tr>
<td></td>
<td>SENTENCE</td>
<td>1247</td>
</tr>
<tr>
<td></td>
<td>OUTCOME</td>
<td>1175</td>
</tr>
<tr>
<td></td>
<td>UNSPECIFIED</td>
<td>759</td>
</tr>
<tr>
<td></td>
<td>CHALLENGE</td>
<td>351</td>
</tr>
<tr>
<td></td>
<td>OUTCOME_CONTRIBUTION</td>
<td>219</td>
</tr>
<tr>
<td></td>
<td>FUTUREWORK</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>CHALLENGE_HYPOTHESIS</td>
<td>7</td>
</tr>
<tr>
<td>MatSci PubMed</td>
<td>RESULTS</td>
<td>1282</td>
</tr>
<tr>
<td></td>
<td>OBJECTIVE</td>
<td>1264</td>
</tr>
<tr>
<td></td>
<td>METHODS</td>
<td>1198</td>
</tr>
<tr>
<td></td>
<td>CONCLUSION</td>
<td>380</td>
</tr>
<tr>
<td></td>
<td>BACKGROUND</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 11: Label counts for the different AZ corpora.

- **BACKGROUND**: For photocatalytic water splitting using photoelectrochemical cells (PECs), the charge carriers are created from the photovoltaic effect close to the catalytic site.

- **PRIOR WORK**: Proton exchange membrane (PEM) electrolysis, which occurs in acidic electrolytes (pH 0–7), has better efficiency and enhanced ramping capability over other types of electrolysis [7].

- **EXPERIMENT**: In order to find an optimum efficiency of the PV–electrolysis, different combinations of the electrolyzer with A-CIGS-based thin film solar cell modules with different band gaps of the cell were examined.

- **PREPARATION**: Pre-sputtering was performed for 5 min in argon plasma in order to remove surface impurities.

- **CHARACTERIZATION**: The current density-potential (j–V) characteristics of the A-CIGS cells were recorded under simulated AM 1.5G sunlight in a set-up with a halogen lamp (ELH).

- **EXPLANATION**: A possible explanation for the superior ECSA-specific activity in the 3D WP-structured catalysts is efficient removal of oxygen bubbles from the catalyst layer.

- **RESULTS**: The load curves were similar for the electrolyzers with different WO3 thin films and the lowest potential needed for 10 mA cm-2 in the overall reaction was 1.77 V.

- **CONCLUSION**: The Cu-N- rGO demonstrated superior catalytic activity to the counterpart N-rGO, and enhanced durability compared to commercial Pt/C.

Structural tags are used, for example, in the following cases.

- **HEADING**: 4. Discussion and concluding remarks

- **METADATA**: This research was funded by Hubei Superior and Distinctive Discipline Group of “Mechatronics and Automobiles” (No.XKQ2019009).

- **CAPTION**: Figure 8. Enlarged view of the shaded portion of Figure 7.

### F.2 Multi-Label Examples

In contrast to earlier works on AZ, our approach to labeling AZ in materials science publications uses a multi-label approach. In this section, we discuss some multi-label examples.

- **BACKGROUND, PRIOR WORK, RESULTS**: This indicates that the HER follows a rate-determining Volmer or Heyrovsky step for different sputtering conditions without any order [40,41]. In this example, a result obtained in the current paper confirms a result known from prior work.

- **EXPERIMENT, CHARACTERIZATION, RESULTS, EXPLANATION**: Attributing this enthalpy release exclusively to the removal of grain boundaries in stage B, a specific grain boundary energy $\gamma = H_{\rho 3d ini-1-dfin}=0.85\pm0.08$fJm-2ls estimated using the initial and final crystallite diameters of stage B, as given above (dini=222nm, dfin=764nm),
as well as the Cu bulk value of 8.92gcm-3 for the mass density ρ. The first subordinate clause of this sentence (Attributing ... stage B) is an EXPLANATION. The remainder of the sentence states a CHARACTERIZATION.

- BACKGROUND, PRIOR WORK, RESULTS, CONCLUSION: Furthermore, the fatigue life decreased approximately by more than 12% when the pre-corroded time was doubled, and the fatigue life decreased approximately by more than 11% when the applied stress level was doubled, indicating that both pre-corroded time and applied stress level can significantly affect the fatigue life of specimens, which shows a good agreement with the previous works [37,38]. This example illustrates a case where our simplification of labeling entire sentences comes to its limits: The first part of the sentence (Furthermore ... was doubled) reports RESULTS while the second part draws a CONCLUSION drawing connections to specific PRIOR WORK.

G Detailed Results

In this section, we provide detailed results for the experiments presented in the main part of the paper.

Table 13 (no oversampling and ML-ROS) and Table 14 (multi-task AZ-CL) show the results in terms of precision, recall and (hierarchical) F1 for each label individually. We report the results on both dev and test of the specific model that performed best on dev compared to all other models.

First, we compare the difference between no oversampling at all and when using ML-ROS. As shown in Table 1, MOTIVATION, METADATA, and CAPTION are the least frequent labels in our dataset. Except for METADATA on the test set, there is always an increase in terms of F1-score when applying ML-ROS on minority labels during training. The biggest increase of 5.8 happened for MOTIVATION on the test set. Furthermore, there is also an improvement of 1.2 points on dev and 2.5 points on test in terms of F1-score for EXPLANATION, which is fourth in the list of rarest AZ labels.

During our experimentation, we observed that ML-ROS tends to be especially helpful for models that show strong performance on majority labels, but not on minority labels. Other models with different hyperparameters achieve even better scores on minority labels without oversampling; however, they tend to have worse overall performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>LM</th>
<th>mic.-F1</th>
<th>mac.-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Oversampling</td>
<td>BERT</td>
<td>72.6±1.0</td>
<td>66.5±0.7</td>
</tr>
<tr>
<td></td>
<td>MatSciBERT</td>
<td>76.3±0.7</td>
<td>70.1±0.7</td>
</tr>
<tr>
<td></td>
<td>SciBERT</td>
<td>76.2±0.9</td>
<td>70.2±0.6</td>
</tr>
<tr>
<td>ML-ROS</td>
<td>SciBERT</td>
<td>76.7±0.7</td>
<td>70.6±0.9</td>
</tr>
<tr>
<td>+ MT (+PM)</td>
<td>SciBERT</td>
<td>76.5±0.4</td>
<td>69.5±0.5</td>
</tr>
<tr>
<td>+ MT (+ART)</td>
<td>SciBERT</td>
<td>75.0±0.9</td>
<td>68.9±1.1</td>
</tr>
<tr>
<td>+ MT (+AZ-CL)</td>
<td>SciBERT</td>
<td>77.2±0.3</td>
<td>71.1±0.5</td>
</tr>
<tr>
<td>+ MT (+DRI)</td>
<td>SciBERT</td>
<td>76.6±0.3</td>
<td>70.5±0.4</td>
</tr>
<tr>
<td>+ MT (+ART+AZ+DRI)</td>
<td>SciBERT</td>
<td>76.4±0.6</td>
<td>70.2±0.5</td>
</tr>
<tr>
<td>Data Augm. (+PM)</td>
<td>SciBERT</td>
<td>77.1±0.8</td>
<td>70.8±1.3</td>
</tr>
</tbody>
</table>

Table 12: Results on MuLMS-AZ test set, hierarchical micro/macro F1: MT=Multi-Task models, *not directly comparable.

Next, we describe the effects of multi-task training with the AZ-CL dataset. We also apply ML-ROS to MuLMS-AZ in our multi-task experiments. Both micro-F1 and macro-F1 increase by 0.5 points in terms of micro- and macro-F1 when using multi-tasking instead of ML-ROS only. Most of the per-label F1-scores increased when using multi-tasking, interestingly with notable differences for CHARACTERIZATION (4.8) and META- DATA (5.6). We conclude that multi-tasking with AZ-CL helps supporting common majority labels even though the domain of this dataset is clearly different from ours.

In contrast, multi-task learning with the other datasets consistently resulted in decreases of performance. The chemistry domain is intuitively closest to that of materials science, hence, we would have expected ART to be a good additional dataset in multi-task learning. Brack et al. (2022) provide some insights into cross-domain learning of AZ categories using datasets from biomedicine, chemistry, and computer graphics. Our MuLMS-AZ, alongside AZ-CL, opens up new research opportunities.

In addition, we perform a data augmentation experiment using AZ data from scientific abstracts of the PubMed Medline corpus11, filtering for abstracts that were published in journals related to the materials science domain (see Appendix C). We map the four PubMed AZ labels BACKGROUND, OBJECTIVE, RESULTS, and CONCLUSIONS to our four AZ labels BACKGROUND, MOTIVATION, RESULTS and CONCLUSION. Augmenting with data from the PubMed Medline dataset also helps to

---

achieve better performance. However, the micro-F1 score is 0.1 lower and the macro-F1 score is 0.3 lower compared to the MT (+AZ-CL) model. On the other hand, training is much more time-efficient since a low augmentation percentage of 10% is sufficient to get good results.
Table 13: Per label scores on dev and test of MuLMS-AZ in terms of precision, recall, and hierarchical F1. **Bold:** best result for label. P, R, and F1 scores are averages over the P, R, F1 scores of 5 folds each.

<table>
<thead>
<tr>
<th>Label</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTIVATION</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>BACKGROUND</td>
<td>62.7</td>
<td>54.1</td>
</tr>
<tr>
<td>-PRIOR WORK</td>
<td>62.7</td>
<td>54.1</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>95.4</td>
<td>84.2</td>
</tr>
<tr>
<td>-CHARACTERIZATION</td>
<td>58.0</td>
<td>69.3</td>
</tr>
<tr>
<td>PREPARATION</td>
<td>69.4</td>
<td>55.6</td>
</tr>
<tr>
<td>RESULTS</td>
<td>77.6</td>
<td>84.4</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>61.8</td>
<td>49.6</td>
</tr>
<tr>
<td>Caption</td>
<td>90.5</td>
<td>77.6</td>
</tr>
<tr>
<td>Heading</td>
<td>84.7</td>
<td>97.7</td>
</tr>
<tr>
<td>Metadata</td>
<td>84.3</td>
<td>72.0</td>
</tr>
<tr>
<td>Average</td>
<td>76.6</td>
<td>70.4</td>
</tr>
<tr>
<td><strong>SciBERT, no oversampling</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOTIVATION</td>
<td>65.5</td>
<td>46.8</td>
</tr>
<tr>
<td>BACKGROUND</td>
<td>89.2</td>
<td>80.0</td>
</tr>
<tr>
<td>-PRIOR WORK</td>
<td>97.0</td>
<td>84.5</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>82.1</td>
<td>85.8</td>
</tr>
<tr>
<td>-CHARACTERIZATION</td>
<td>72.0</td>
<td>68.9</td>
</tr>
<tr>
<td>PREPARATION</td>
<td>65.2</td>
<td>65.1</td>
</tr>
<tr>
<td>RESULTS</td>
<td>46.3</td>
<td>33.0</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>56.7</td>
<td>55.3</td>
</tr>
<tr>
<td>Caption</td>
<td>92.4</td>
<td>75.2</td>
</tr>
<tr>
<td>Heading</td>
<td>84.8</td>
<td>97.9</td>
</tr>
<tr>
<td>Metadata</td>
<td>93.1</td>
<td>68.0</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>76.6</td>
<td>70.4</td>
</tr>
<tr>
<td><strong>SciBERT, ML-ROS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOTIVATION</td>
<td>56.3</td>
<td>55.9</td>
</tr>
<tr>
<td>BACKGROUND</td>
<td>82.2</td>
<td>84.8</td>
</tr>
<tr>
<td>-PRIOR WORK</td>
<td>96.0</td>
<td>84.5</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>85.1</td>
<td>83.2</td>
</tr>
<tr>
<td>-CHARACTERIZATION</td>
<td>73.3</td>
<td>67.3</td>
</tr>
<tr>
<td>PREPARATION</td>
<td>69.4</td>
<td>63.4</td>
</tr>
<tr>
<td>RESULTS</td>
<td>45.7</td>
<td>35.2</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>60.6</td>
<td>44.5</td>
</tr>
<tr>
<td>Caption</td>
<td>91.7</td>
<td>79.6</td>
</tr>
<tr>
<td>Heading</td>
<td>85.4</td>
<td>97.5</td>
</tr>
<tr>
<td>Metadata</td>
<td>89.3</td>
<td>70.5</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>76.1</td>
<td><strong>70.8</strong></td>
</tr>
</tbody>
</table>

Table 14: Per label scores on dev and test in terms of precision, recall, and hierarchical F1 using multi-task learning with the AZ-CL dataset, SciBERT, ML-ROS.

```text
<table>
<thead>
<tr>
<th>Label</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOTIVATION</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>BACKGROUND</td>
<td>62.7</td>
<td>54.1</td>
</tr>
<tr>
<td>-PRIOR WORK</td>
<td>95.4</td>
<td>84.2</td>
</tr>
<tr>
<td>EXPERIMENT</td>
<td>95.4</td>
<td>84.2</td>
</tr>
<tr>
<td>-CHARACTERIZATION</td>
<td>95.4</td>
<td>84.2</td>
</tr>
<tr>
<td>PREPARATION</td>
<td>83.6</td>
<td>82.8</td>
</tr>
<tr>
<td>RESULTS</td>
<td>42.6</td>
<td>35.8</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>61.8</td>
<td>49.6</td>
</tr>
<tr>
<td>Caption</td>
<td>90.5</td>
<td>77.6</td>
</tr>
<tr>
<td>Heading</td>
<td>84.7</td>
<td>97.7</td>
</tr>
<tr>
<td>Metadata</td>
<td>84.3</td>
<td>72.0</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>76.1</td>
<td><strong>70.8</strong></td>
</tr>
</tbody>
</table>
```

Table 14: Per label scores on dev and test in terms of precision, recall, and hierarchical F1 using multi-task learning with the AZ-CL dataset, SciBERT, ML-ROS.
A Side-by-side Comparison of Transformers for English Implicit Discourse Relation Classification

Bruce W. Lee1,2, BongSeok Yang2, Jason Hyeong-Jong Lee2
1University of Pennsylvania - PA, USA
2LXPER AI Research - Seoul, South Korea
brucelws@seas.upenn.edu
bongseok@lxper.com
jasonlee@lxper.com

Abstract

Though discourse parsing can help multiple NLP fields, there has been no wide language model search done on implicit discourse relation classification. This hinders researchers from fully utilizing public-available models in discourse analysis. This work is a straightforward, fine-tuned discourse performance comparison of seven pre-trained language models. We use PDTB-3, a popular discourse relation annotated dataset. Through our model search, we raise SOTA to 0.671 ACC and obtain novel observations. Some are contrary to what has been reported before (Shi and Demberg, 2019b), that sentence-level pre-training objectives (NSP, SBO, SOP) generally fail to produce the best performing model for implicit discourse relation classification. Counterintuitively, similar-sized PLMs with MLM and full attention led to better performance.

1 Introduction

An utterance has multiple dimensions of meaning. Discourse relation classification identifies one such dimension: the coherence relation between clauses or sentences arising from low-level textual cues (Zhao and Webber, 2022; Webber et al., 2019). This makes the task important to several NLP fields, including multi-party dialogue analysis (Li et al., 2022), social media postings analysis (Siskou et al., 2022), and student literary writing analysis (Fiacco et al., 2022). A discourse relation is often marked with explicit connectives such as but, because, and.

Consider the following example:

"Although Philip Morris typically tries to defend the rights of smokers, ["this has nothing to do with cigarettes, nor will it ever," the spokesman says]Arg1, [But]Conn [some anti-smoking activists disagree]Arg2, expressing anger... → Comparison.Contrast"

The explicit connective, Conn (But), is informative. Hence, it is fairly easy to know that the two arguments, Arg1 and Arg2, are compared, likely in a contrasting relationship rather than similarity. This task is often referred to as explicit discourse relation classification. Pitler and Nenkova (2009) achieves a 94.15% accuracy (4-way) with Naive Bayes.

Implicit discourse relation classification, on the other hand, aims to classify discourse relationships in cases without an explicit connective. It has received constant attention (Li et al., 2022) since the release of Penn Discourse Tree Bank 2.0 (PDTB-2) (Prasad et al., 2008). Consider the following:

"Last year we probably bought one out of every three new deals.\Arg1," he says. "[This year, at best, it’s in one in every five or six.]Arg2" → Comparison.Contrast

Without an explicit connective, Conn, discourse relation classification only relies on low-level semantic cues from the arguments, Arg1 and Arg2. Such “implicit” discourse relation classification is very challenging as it requires a language model to conceptualize the unstated goal the speaker is trying to achieve, not only the literal content (Shi and Demberg, 2019b; Sileo et al., 2019).

With XLNet\textsubscript{large} (Yang et al., 2019) achieving ~60% accuracy (Kim et al., 2020), pre-trained language models showed promising improvements from the past studies: Maximum-Entropy Learning (~40% F1) (Lin et al., 2014), Adversarial Network (~46% ACC) (Qin et al., 2017), Seq2Seq + Memory Network (~48% ACC) (Shi and Demberg, 2019a). Implicit discourse relation classification gives relatively small textual information for a language model to infer from. Thus, pre-training large text helps establish typical relations within/across clauses and sentences (Shi and Demberg, 2019b).
Pre-trained language models, like BERT (Devlin et al., 2018), follow transformer-type (Vaswani et al., 2017) architecture and have only been recently introduced into implicit discourse relation classification (Kishimoto et al., 2020). To the best of our knowledge, BERT and XLNet are the only pre-trained language models (fine-tuned and) evaluated for implicit discourse relation classification on PDTB-3 (Kim et al., 2020). However, language models vary in architecture, training objective, data, etc.

Instead of performing a focused study on a single model, we fine-tune seven state-of-the-art (SOTA) language models (§2). Our wider approach brings weaknesses (§5) (as we ignore some model-specific characteristics), but it allows the bird’s-eye view of several downstream performances in PDTB-3 (§3) (Webber et al., 2019) and raises SOTA (∼67% ACC) on Kim et al. (2020)’s evaluation protocol. By contrasting performances, we show that certain language model characteristics can benefit implicit discourse relation classification.

Additionally, we take the best-performing language model and check if the "full-sentence(s)" setup gives better performance (§3.4). As we elaborate further in the following sections, our sanity checks on PDTB-3 hint that some argument annotations are questionable in terms of consistency and coverage. Hence, implicit discourse relation classification accuracy might improve by simply training the language model with a full sentence(s) instead of human-annotated argument spans (Arg1 and Arg2). We evaluate this idea toward the end.

Table 1: Tested language models and their varying configurations. ∗: BART follows the original encoder-decoder architecture, 12 layers allocated for each. **: DeBERTa uses disentangled attention. MLM: masked language modelling. SOP: sentence order prediction. SBO: span boundary objective. TI: text infilling. SS: sentence shuffling.

<table>
<thead>
<tr>
<th>Configurations</th>
<th>ALBERT&lt;sub&gt;large&lt;/sub&gt;</th>
<th>BART&lt;sub&gt;large&lt;/sub&gt;</th>
<th>BigBird-R.</th>
<th>DeBERTa&lt;sub&gt;large&lt;/sub&gt;</th>
<th>Longformer&lt;sub&gt;large&lt;/sub&gt;</th>
<th>RoBERTa&lt;sub&gt;large&lt;/sub&gt;</th>
<th>SpanBERT&lt;sub&gt;large&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>17M</td>
<td>406M</td>
<td>-</td>
<td>350M</td>
<td>435M</td>
<td>340M</td>
<td>340M</td>
</tr>
<tr>
<td>Hidden</td>
<td>1024</td>
<td>1024</td>
<td>-</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Layers</td>
<td>24 (Enc)</td>
<td>24 (Enc+Dec)∗</td>
<td>-</td>
<td>24 (Enc)</td>
<td>24 (Enc)</td>
<td>24 (Enc)</td>
<td>24 (Enc)</td>
</tr>
<tr>
<td>Attention Heads</td>
<td>16</td>
<td>16</td>
<td>-</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Self-Attention</td>
<td>Full</td>
<td>Full</td>
<td>Block-Sparse</td>
<td>Full**</td>
<td>Global+Window</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Max Seq. Length</td>
<td>512</td>
<td>512</td>
<td>4096</td>
<td>512</td>
<td>4096</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>Pre-train Obj.</td>
<td>MLM &amp; SOP</td>
<td>TI &amp; SS</td>
<td>-</td>
<td>MLM</td>
<td>MLM</td>
<td>MLM</td>
<td>MLM &amp; SBO</td>
</tr>
</tbody>
</table>

Pre-trained language model’s competence in discourse was questionable until Shi and Demberg (2019b) proposed that BERT’s pre-training objective can benefit implicit discourse relation classification. However, Iter et al. (2020) hints that BERT is not the language model best suited to the task.

Implicit discourse relation classification is an active area of research (Kurfah, 2022; Zhao and Webber, 2022; Kurfah and Östling, 2021b; Knaebel, 2021; Munir et al., 2021; Kurfah and Östling, 2021a; Kishimoto et al., 2020; Bourgonje and Stede, 2019; Shi and Demberg, 2019b; Bai and Zhao, 2018; Dai and Huang, 2018; Rutherford et al., 2017). However, there has been no wide-range model study on implicit discourse relation classification, limiting a researcher’s scope of model choice. This issue is further complicated by the fact that discourse task performances do not always correlate with popular semantics-based natural language understanding (NLU) scores, such as GLUE (Sileo et al., 2019). Thus, it is difficult to predict which language model can perform well without a dedicated empirical exploration.

With the a version update to Penn Discourse Tree Bank (PDTB-3) (Webber et al., 2019) and the correspondingly updated evaluation method (Kim et al., 2020), we fine-tune seven language models to implicit discourse relation classification.

The chosen language models are: RoBERTa<sub>large</sub> (Liu et al., 2019), ALBERT<sub>large</sub> (Lan et al., 2019), BigBird-RoBERTa<sub>large</sub> (Zaheer et al., 2020), BART<sub>large</sub> (Lewis et al., 2020), Longformer<sub>large</sub> (Beltagy et al., 2020), SpanBERT<sub>large</sub> (Joshi et al., 2020), DeBERTa<sub>large</sub> (He et al., 2020a). These models are selected with diversity in mind, especially in terms of input sequence length, attention type, and pre-train objectives. These models fol-
low the popular transformer architecture (Vaswani et al., 2017), and we will not review each model in detail. A brief comparison is shown in Table 1.

### 3 Experiments

#### 3.1 Data Preparation

We obtained the official PDTB-3 data from the Linguistic Data Consortium\(^1\). PDTB-3 is a large-scale resource of annotated discourse relations and their arguments over the 1 million words Wall Street Journal Corpus (Marcus et al., 1993). From a public repository\(^2\), we retrieved the corresponding evaluation script (Kim et al., 2020). We describe some characteristics of the evaluation protocol below.

**Cross-validation** is used on the section level to preserve paragraph and document structures. Cross-validation likely solves label sparsity issue (Shi and Demberg, 2017). The 25 sections of PDTB-3 are divided into 12 folds with 2 development, 2 test, and 21 training sections in each fold. The sliding window of two sections is used, creating 12 folds.

**Label set** is composed of 14 senses on L2 discourse relations (see Appendix B). Only the senses with \(\geq 100\) instances are used. This is to produce results that are in align with Kim et al. (2020). This alignment is crucial as we directly compared our results against fine-tuned BERT from Kim et al. (2020), which is trained with next sentence prediction (NSP) objective. Multiply-annotated labels become separate training instances.

#### 3.2 Fine-Tuning

To ensure reproducibility, we only take pre-trained language models from the now ubiquitous Huggingface (Wolf et al., 2019) transformers library. Fine-tuning was done with PyTorch (Paszke et al., 2019) and our scripts are publicly available.

During fine-tuning, each training instance is a concatenation of two arguments (sequence of tokens in \(\text{Arg}_1\) and \(\text{Arg}_2\)). BERT-type models carry special tokens ([CLS], [SEP], [EOS]) for segmentation: [CLS], \(\text{Arg}_1\_1, \ldots, \text{Arg}_1\_N,\) [SEP], \(\text{Arg}_2\_1, \ldots, \text{Arg}_2\_M,\) [EOS]. Depending on the model, these special tokens are modified or completely removed.

As for hyperparameter searches, we mostly focus on the learning rate. We use the popular AdamW optimizer with a linear scheduler (no warm-up steps). As for the learning rate, we start from 2e-5, a value commonly used for text classification since Sun et al. (2019). We test lower learning rates of 2e-6 and 5e-6; we find that 5e-6 (which is slightly lower than what is usually used in sequence classification) performs best for almost all models. The batch size is 8 and the max input length is set at 256.

Lastly, for each experiment step (i.e. BART on fold 1), we train for 10 epochs with an early stop. The training stops if the current epoch’s validation loss (see development set §3.1) did not decrease from the previous epoch. Model training time, GPU, language model repository address, and other details on hyperparameters are in Appendix C.

#### 3.3 Evaluation and Observations

In Table 2-a, we report the mean test set accuracy of 12 folds along with variance. This is in alignment with what was recommended by Kim et al. (2020). Development set performances are given in Table 3 to facilitate reproducibility. For multiply-annotated labels (also discussed in §3.1), the model only has to get one label correct. We reach some surprising observations, which we share below.

1) **Sentence-level pre-train objectives** are not necessary to create best-performing models. During fine-tuning, each training instance is a concatenation of two arguments (sequence of tokens in \(\text{Arg}_1\) and \(\text{Arg}_2\)). BERT-type models carry special tokens ([CLS], [SEP], [EOS]) for segmentation: [CLS], \(\text{Arg}_1\_1, \ldots, \text{Arg}_1\_N,\) [SEP], \(\text{Arg}_2\_1, \ldots, \text{Arg}_2\_M,\) [EOS]. Depending on the model, these special tokens are modified or completely removed.

2) **Cross-validation** is used on the section level to preserve paragraph and document structures. Cross-validation likely solves label sparsity issue (Shi and Demberg, 2017). The 25 sections of PDTB-3 are divided into 12 folds with 2 development, 2 test, and 21 training sections in each fold. The sliding window of two sections is used, creating 12 folds.

3) **Label set** is composed of 14 senses on L2 discourse relations (see Appendix B). Only the senses with \(\geq 100\) instances are used. This is to produce results that are in alignment with Kim et al. (2020). This alignment is crucial as we directly compared our results against fine-tuned BERT from Kim et al. (2020), which is trained with next sentence prediction (NSP) objective. Multiply-annotated labels become separate training instances.

### Table 2: Language model performances (test set) on Level-2 14-way implicit discourse relation classification.

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>ALBERT(_{\text{large}})</th>
<th>BART(_{\text{large}})</th>
<th>BigBird-R</th>
<th>DeBERTa(_{\text{large}})</th>
<th>Longformer(_{\text{large}})</th>
<th>RoBERTa(_{\text{large}})</th>
<th>SpanBERT(_{\text{large}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>5e-6</td>
<td>5e-6</td>
<td>5e-6</td>
<td>2e-6</td>
<td>5e-6</td>
<td>2e-6</td>
<td>5e-6</td>
</tr>
<tr>
<td>a: Argument Spans</td>
<td>Accuracy</td>
<td>0.565</td>
<td>0.657</td>
<td>0.649</td>
<td>0.671</td>
<td>0.668</td>
<td>0.670</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>2.53e-4</td>
<td>2.15e-4</td>
<td>4.02e-4</td>
<td>2.70e-4</td>
<td>2.15e-4</td>
<td>3.32e-4</td>
</tr>
<tr>
<td>b: Full Sentence(s)</td>
<td>Accuracy</td>
<td>0.534</td>
<td>0.629</td>
<td>0.620</td>
<td>0.634</td>
<td>0.627</td>
<td>0.617</td>
</tr>
<tr>
<td></td>
<td>Variance</td>
<td>2.27e-4</td>
<td>4.28e-4</td>
<td>2.79e-4</td>
<td>3.75e-4</td>
<td>4.18e-4</td>
<td>3.62e-4</td>
</tr>
</tbody>
</table>
relation classification after conducting an ablation study on BERT. Their finding was intuitive as well because implicit discourse relation classification aims to find the relationship between two argument spans.

But in a more general scope, the necessity of NSP has been questioned multiple times (Yang et al., 2019; Lample and Conneau, 2019). In other words, NSP – or any other sentence-level pre-train objective for that matter – could have been only helpful in some specific ablation study of BERT-type models but not in other cases (Liu et al., 2019). We obtain supporting results in Table 2-a, where language models with sentence-level objectives performed worse than MLM-only models given similar model sizes (ALBERT is an exception).

2) Long-document modifications (mostly done by altering attention schemes of an existing model) decrease the original model performance. At first, we postulated that long-document models could lead to performance increases because they can learn long-span discourse relations during pre-training. But using sparse or block attention mechanisms eventually led to a performance decrease.

The decrease is clearly demonstrated by BigBird-RoBERTa<sub>large</sub> and Longformer<sub>large</sub>. Both models start from the existing RoBERTa<sub>large</sub> checkpoint and modify it to process longer sequences. Such modifications achieved performance increases in other NLP tasks like question-answering, coreference resolution, and some cases of sequence classification. But implicit discourse relation classification, which requires the model’s understanding of dense discourse relations hidden within a few tokens, long-document modification is a drawback.

3) The simplest combination of MLM and full attention is best suited for implicit discourse relation classification. We are making this argument within the scope of what we have tested. We believe that MLM and full attention (e.g., RoBERTa, DeBERTa) work best because the model has to make inferences based on a relatively small number of tokens. Hence, trivial textual cues should not be risked being overlooked. MLM, with full attention, forces every token to attend to every other and learn the token-specific relations, likely to lose the least textual cues and nuances.

3.4 Train Full Sentence or Argument Span?
Following the aforementioned observations, we postulated that fine-tuning language models using full sentence(s) could further improve classification accuracy. By full sentence(s), we refer to the sentence(s) (usually up to two) that the annotated argument spans appeared. We had two reasons for our postulation: 1. textual cues that hint at underlying discourse relation could be spread throughout the sentence(s), 2. argument span annotation is sometimes inconsistent, especially at punctuation marks, unnecessarily confusing the language model. Implicit discourse relation classification has rarely been tested using the full sentence.

We built an argument matcher to find the source sentence of each annotated argument span. For inter-sentential relations, we only considered argument spans that came from two adjacent source sentences. We share the test set results in Table 2-b. The results bring us to our fourth observation.

4) As input, concatenating argument spans generally perform better than full sentence(s). Opposed to our postulation, using full sentence(s) as input decreased performance on the test set. Though we see mixed results on the development set in Table 3, training full sentences as input generally decrease performance. But when it comes to implicit discourse sense classification from the raw text (that means in practical, end-to-end applications), the benefits of using argument spans must be weighed against the low accuracies (50% ~ 60%) of the available argument extractors.

4 Conclusion
Researchers often build or modify a neural network to improve task performance. While such effort is essential, this paper shows that SOTA can also be raised through extensive search and application of
existing resources. Through a side-by-side comparison of seven PLMs, we also make handy observations on pre-training objectives, long-document modifications, and full-sentence setups. Though some might consider these phenomena rather expected, nothing is scientifically conclusive until an analysis is performed at an adequate scale. We hope that our report helps researchers working towards discourse understanding, and we continue to discuss the missing details in the appendices.

5 Acknowledgement

We thank the anonymous reviewers for their crisp and realistic advices on cleaning the language of the paper and experiments. Most of the review opinions were accepted and were reflected in the paper as they were valid.

References


the Association for Computational Linguistics, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.


A “Full sentence(s)” Experiment

A.1 What Makes the Experiment Important?

This section is a continuation of §3.4. Here, we discuss implicit discourse relation classification from raw sentence(s), which we believe is the best practical example of real-world applications of the related fields. Such an end-to-end concept has been popularized through CoNLL-2016 (Xue et al., 2016) and CoNLL-2015 (Xue et al., 2015), and most systems develop a separate argument span identification model. Then, the identified argument spans would be fed to the discourse relation classification model for sense prediction (refer to examples given in §1) (He et al., 2020b).

Such a double-step process makes sense. Indeed, feeding the exact argument spans (that only contain the tokens that imply a certain discourse sense) will increase sense prediction performance.

But the problem arises because identifying argument spans from raw sentence(s) is a low accuracy operation (Knaebel, 2021). A wrong span identification eventually leads to error propagation, providing a discourse relation classification model that lacks textual information. We give a theoretical error propagation example and conduct a simple experiment to prove our point.

A.2 Theoretical Example of Error Propagation

1. A set of two raw sentences is given.

"Last year we probably bought one out of every three new deals," he says. "This year, at best, it’s in one in every five or six."

2. Where correct argument spans are as below.

["Last year we probably bought one out of every three new deals," Arg1 he says. "This year, at best, it’s in one in every five or six.

3. But an argument span identification model often makes wrong predictions (best system (?) at CoNLL-2016 scores 52.02 F1, for exact span match).

["Last year we probably bought one out of every three new deals," Arg1 he says. "This year, at best, it’s in one in every five or six.

4. Now, compare the amount of textual information passed over to the implicit discourse relation classification model, under three setups. Note that setup 1 cannot be used in real-world settings because it requires PDTB-3’s gold annotations.

Setup 1) PDTB-3 (with gold annotations)

Last year we probably bought one out of every three new deals This year, at best, it’s in one in every five or six.

Setup 2) A low accuracy argument span model

Last year we probably bought one it’s in one in every five or six.
Table 4: BERT’s performance (12-folds test set) on PDTB-3’s argument spans.

<table>
<thead>
<tr>
<th>Fine-tuned PLM</th>
<th>Argument Span</th>
<th>ACC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT&lt;sub&gt;large&lt;/sub&gt;</td>
<td>0.912</td>
<td>0.742</td>
<td></td>
</tr>
</tbody>
</table>

Setup 3) Full sentence(s)

"Last year we probably bought one out of every three new deals," he says. "This year, at best, it's in one in every five or six."

A.3 Experiment on Error Propagation

Though not all tokens are valuable under a full sentence(s) setup, we can notice that it is a foolproof way to input all meaningful tokens. Table 4 reports the classification performance of BERT<sub>large</sub>, which was trained to identify argument spans using PDTB-3. Our argument span scoring scheme approximately matches CoNLL-16’s partial scoring scheme, essentially a relaxed version of conlleval. That means we consider a prediction correct if more than 70% of argument span tokens are identified. For implicit discourse relation classification, a sense prediction is correct if it matches any of the multiply-annotated senses.

BERT’s 0.912 ACC score implies that the model could correctly identify at least 70% of the gold argument span tokens more than 9 out of 10 times. Nonetheless, error propagation detrimentally affected implicit discourse relation classification performance in Table 5. This empirically proves our ideas in Appendix A.1.

<table>
<thead>
<tr>
<th>Fine-tuned PLM</th>
<th>Implicit Sense</th>
<th>ACC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeBERT&lt;sub&gt;large&lt;/sub&gt;</td>
<td>0.670</td>
<td>0.671</td>
<td></td>
</tr>
<tr>
<td>with error propagation</td>
<td>0.476</td>
<td>0.491</td>
<td></td>
</tr>
<tr>
<td>full sentence(s)</td>
<td>0.634</td>
<td>0.637</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: DeBERTa performances (12-fold test set) on PDTB-3’s Level-2 14-way implicit discourse relation classification, but under three different pipeline setups.

early stop. The performances reported in Table 2 are obtained with early stop.

ALBERT<sub>large</sub>
- `huggingface.co/albert-large-v1`
- ∼2.4 days, for 12 folds ×10 epochs

BART<sub>large</sub>
- `huggingface.co/facebook/bart-large`
- ∼3.6 days, for 12 folds ×10 epochs

BigBird-RoBERTa<sub>large</sub>
- `huggingface.co/google/bigbird-roberta-large`
- ∼3.2 days, for 12 folds ×10 epochs

DeBERTa<sub>large</sub>
- `huggingface.co/microsoft/deberta-large`
- ∼4.6 days, for 12 folds ×10 epochs

Longformer<sub>large</sub>
- `huggingface.co/allenai/longformer-large-4096`
- ∼11 days, for 12 folds ×10 epochs

RoBERTa<sub>large</sub>
- `huggingface.co/roberta-large`
- ∼2.9 days, for 12 folds ×10 epochs

SpanBERT<sub>large</sub>
- `.../SpanBERT/spanbert-large-cased`
- ∼2.9 days, for 12 folds ×10 epochs

Table 6: Counts of 14-way implicit discourse senses.

Label | Counts
---|---
Comparison.Concession | 1494
Comparison.Contrast | 983
Contingency.Cause | 5785
Contingency.Cause+Belief | 202
Contingency.Condition | 199
Contingency.Purpose | 1373
Expansion.Conjunction | 4386
Expansion.Equivalence | 336
Expansion.Instantiation | 1533
Expansion.Level-of-detail | 3361
Expansion.Manner | 739
Expansion.Substitution | 450
Temporal.Asynchronous | 1289
Temporal.Synchronous | 539

B 14-way Label Set

C More on Fine-tuning Set Up

We ran all our experiments on a single NVIDIA Tesla V100 GPU. Model train time and repositories are listed below. Training times below suppose no
Ensemble Transfer Learning for Multilingual Coreference Resolution

Tuan Lai, Heng Ji
Department of Computer Science
University of Illinois Urbana-Champaign
{tuanml2, hengji}@illinois.edu

Abstract

Entity coreference resolution is an important research problem with many applications, including information extraction and question answering. Coreference resolution for English has been studied extensively. However, there is relatively little work for other languages. A problem that frequently occurs when working with a non-English language is the scarcity of annotated training data. To overcome this challenge, we design a simple but effective ensemble-based framework that combines various transfer learning (TL) techniques. We first train several models using different TL methods. Then, during inference, we compute the unweighted average scores of the models’ predictions to extract the final set of predicted clusters. Furthermore, we also propose a low-cost TL method that bootstraps coreference resolution models by utilizing Wikipedia anchor texts. Leveraging the idea that the coreferential links naturally exist between anchor texts pointing to the same article, our method builds a sizeable distantly-supervised dataset for the target language that consists of tens of thousands of documents. We can pre-train a model on the pseudo-labeled dataset before finetuning it on the final target dataset. Experimental results on two benchmark datasets, OntoNotes and SemEval, confirm the effectiveness of our methods. Our best ensembles consistently outperform the baseline approach of simple training by up to 7.68% in the F1 score. These ensembles also achieve new state-of-the-art results for three languages: Arabic, Dutch, and Spanish.\(^1\)

1 Introduction

Within-document entity coreference resolution is the process of clustering entity mentions in a document that refer to the same entities (Ji et al., 2005; Luo and Zitouni, 2005; Ng, 2010, 2017). It is an important research problem, with applications in various downstream tasks such as entity linking (Ling et al., 2015; Kundu et al., 2018), question answering (Dhingra et al., 2018), and dialog systems (Gao et al., 2019). Researchers have recently proposed many neural methods for coreference resolution, ranging from span-based end-to-end models (Lee et al., 2017, 2018) to formulating the task as a question answering problem (Wu et al., 2020b). Given enough annotated training data, deep neural networks can learn to extract useful features automatically. As a result, on English benchmarks with abundant labeled training documents, the mentioned neural methods consistently outperform previous handcrafted feature-based techniques (Raghu-nathan et al., 2010; Lee et al., 2013), achieving new state-of-the-art (SOTA) results.

Compared to the amount of research on English coreference resolution, there is relatively little work for other languages. A problem that frequently occurs when working with a non-English language is the scarcity of annotated training data. For example, the benchmark OntoNotes dataset contains about eight times more documents in English than in Arabic (Pradhan et al., 2012). Some recent studies aim to overcome this challenge by applying standard cross-lingual transfer learning (TL) methods such as continued training or joint training (Kundu et al., 2018; Pražák et al., 2021). In continued training, a model pretrained on a source dataset is further finetuned on the final target dataset. Experimental results on two benchmark datasets, OntoNotes and SemEval, confirm the effectiveness of our methods. Our best ensembles consistently outperform the baseline approach of simple training by up to 7.68% in the F1 score. These ensembles also achieve new state-of-the-art results for three languages: Arabic, Dutch, and Spanish.\(^1\)

\(^1\)Data and code will be made available upon publication.
Figure 1: An overview of our framework. We first train several coreference resolution models using different TL approaches. During inference, we use a simple unweighted averaging method to combine the models’ predictions.

In summary, our main contributions include:
- We introduce an ensemble-based framework that combines various TL methods effectively.
- We design a new TL method that leverages Wikipedia to bootstrap coreference models.
- Extensive experimental results show that our proposed methods are highly effective and provide useful insights into entity coreference resolution for non-English languages.

2 Methods

Figure 1 shows an overview of our framework. During the training stage, we train several coreference resolution models using various TL approaches. For simplicity, we use the same span-based architecture (Section 2.1) for every model to be trained. However, starting from the same architecture, using different learning methods typically results in models with different parameters. In this work, our framework uses two types of TL methods: (a) cross-lingual TL approaches (Section 2.2) and (b) our newly proposed Wikipedia-based approach (Section 2.3). The cross-lingual TL methods require a labeled coreference resolution dataset in some source language (usually English).

In this work, we propose an effective ensemble-based framework for combining various TL techniques. We first train several coreference models using different TL methods. During inference, we compute the unweighted average scores of the models’ predictions to extract the final set of mention clusters. We also propose a low-cost TL method that bootstraps coreference models without using a labeled dataset in some source language. The basic idea is that the coreference relation often holds between anchor texts pointing to the same Wikipedia article. Based on this observation, our TL method builds a sizable distantly-supervised dataset for the target language from Wikipedia. We can then pretrain a model on the pseudo-labeled dataset before finetuning it on the final target dataset. Experimental results on two datasets, OntoNotes and SemEval (Recasens et al., 2010), confirm the effectiveness of our proposed methods. Our best ensembles outperform the baseline approach of simple training by up to 7.68% absolute gain in the F1 score. These ensembles also achieve new SOTA results for three languages: Arabic, Dutch, and Spanish.
source language, but our Wikipedia-based method
does not have that limitation. Our framework is
general as it can work with other learning methods
(e.g., self-distillation). During inference, we use a
simple unweighted averaging method to combine
the trained models’ predictions (Section 2.4).

2.1 Span-based End-to-End Coreference
Resolution

In this work, the architecture of every model is
based on the popular span-based e2e-coref model
(Lee et al., 2017). Given an input document consist-
ing of \( n \) tokens, our model first forms a contextu-
alized representation for each input token using a
multilingual Transformer encoder such as XLM-R
(Conneau et al., 2020). Let \( \mathbf{X} = (x_1, \ldots, x_n) \) be the
output of the encoder. For each candidate span \( i \),
we define its representation \( \mathbf{g}_i \) as:

\[
\mathbf{g}_i = \left[ \mathbf{x}_{\text{START}(i)}, \mathbf{x}_{\text{END}(i)}, \mathbf{x}_{i}, \phi(s_i) \right] \quad (1)
\]

where \( \text{START}(i) \) and \( \text{END}(i) \) denote the start and
end indices of span \( i \) respectively. \( \mathbf{x}_{i} \) is an attention-
weighted sum of the token representations in the
span (Lee et al., 2017). \( \phi(s_i) \) is a feature vector
encoding the size of the span.

To maintain tractability, we only consider spans
with up to \( L \) tokens. The value of \( L \) is selected
empirically and set to be 30. All the span represen-
tations are fed into a mention scorer \( s_m(\cdot) \):

\[
s_m(i) = \text{FFNN}_m(\mathbf{g}_i) \quad (2)
\]

where \( \text{FFNN}_m \) is a feedforward neural network
with ReLU activations. Intuitively, \( s_m(i) \) indicates
whether span \( i \) is indeed an entity mention.

After scoring the spans using \( \text{FFNN}_m \), we only
keep spans with high mention scores\(^2\). We denote
the set of the unpruned spans as \( S \). Then, for each
remaining span \( i \in S \), the model predicts a distri-
bution \( P(j) \) over its antecedents\(^3\) \( j \in Y(i) \):

\[
P(j) = \frac{\exp(s(i,j))}{\sum_{k \in Y(i)} \exp(s(i,k))} \quad (3)
\]

\[
s(i, j) = \text{FFNN}_s(\left[ \mathbf{g}_i, \mathbf{g}_j, \phi(i, j) \right])
\]

where \( Y(i) = \{ \epsilon, 1, \ldots, i - 1 \} \) is a set consisting of
a dummy antecedent \( \epsilon \) and all spans that precede \( i \).

The dummy antecedent \( \epsilon \) represents two possible
cases: (1) the span \( i \) is not an entity mention, or
(2) the span \( i \) is an entity mention, but it is not
coreferential with any remaining preceding span.
\( \text{FFNN}_s \) is a feedforward network, and \( \phi \) is ele-
ment-wise multiplication. \( \phi(i, j) \) encodes the distance
between the two spans \( i \) and \( j \).

Given a labeled document \( D \) and a model with
parameters \( \theta \), we define the mention detection loss:

\[
\mathcal{L}_{\text{detect}}(\theta, D) = -\frac{1}{|S|} \sum_{i \in S} \mathcal{L}_{\text{detect}}(\theta, i)
\]

where \( \mathcal{L}_{\text{detect}}(\theta, i) = y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \)
with \( \hat{y}_i = \text{sigmoid}(s_m(i)) \), and \( y_i = 1 \) if and
only if span \( i \) is in one of the gold-standard mention
clusters. In addition, we also want to maximize the
marginal log-likelihood of all correct antecedents
implied by the gold-standard clustering:

\[
\mathcal{L}_{\text{cluster}}(\theta, D) = -\log \prod_{i \in S} \sum_{g \in Y(i) \cap \text{GOLD}(i)} \hat{P}(\hat{y})
\]

where \( \text{GOLD}(i) \) are gold antecedents for span \( i \).
\( \hat{P}(\hat{y}) \) is calculated using Equation 3. Our final loss
combines mention detection and clustering:

\[
\mathcal{L}(\theta, D) = \mathcal{L}_{\text{detect}}(\theta, D) + \mathcal{L}_{\text{cluster}}(\theta, D) \quad (4)
\]

2.2 Cross-Lingual Transfer Learning

Inspired by previous studies (Xia and Van Durme,
2021; Min, 2021; Pražák et al., 2021), we investi-
gate two different cross-lingual transfer learning
methods: continued training and joint training.
Both methods assume the existence of a labeled
dataset in some source language. In this work, we
use the English OntoNotes dataset (Pradhan et al.,
2012) as the source dataset, as it contains nearly
3,500 annotated documents (Table 1).

Continued Training. We first train a coreference
resolution model on the source dataset until con-
vergence. After that, we further finetune the pre-
trained model on a target dataset. More formally,
let \( M(f, \theta_0) \) denote an optimization procedure for
\( f \) with initial guess \( \theta_0 \). This optimization pro-
duce can, for example, be the application of some
stochastic gradient descent algorithm. Also, let \( S \)
be the set of all training documents in the source
dataset, and let \( T \) denote the set of all training
documents in the target dataset. Then, the first stage of
continued training can be described as:

\[
\hat{\theta}_1 = M \left( \sum_{D \in S} \mathcal{L}(\theta, D), \hat{\theta}_0 \right) \quad (5)
\]
Figure 2: Since the hyperlinks of *Tigre de México* and *Tigres UANL* point to the same Wikipedia page, a person who does not know Spanish can still guess that the two mentions are likely to be coreferential. In fact, the two mentions both refer to Tigres UANL, a Mexican professional football club.

where $\hat{\theta}_0$ is randomly initialized. Then, the second stage can be described using Equation 6:

$$\hat{\theta}_2 = M \left( \sum_{D \in \mathcal{T}} \mathcal{L}(\theta, D), \hat{\theta}_1 \right)$$

Here, $\hat{\theta}_2$ is the parameter set of the final model.

**Joint Training.** We combine both the source and target datasets to train a model. More specifically, using the same notations as above, we can describe joint training by the following equation:

$$\hat{\theta}_1 = M \left( \sum_{D \in \mathcal{T} \cup \mathcal{S}} \mathcal{L}(\theta, D), \hat{\theta}_0 \right)$$

where $\hat{\theta}_0$ is randomly initialized, and $\hat{\theta}_1$ is the parameter set of the final model.

### 2.3 Bootstrapping using Wikipedia Hyperlinks

While cross-lingual methods such as continued training and joint training are conceptually simple and typically effective (Huang et al., 2020), they require the existence of a labeled dataset in some source language. To overcome this limitation, we propose an inexpensive TL method that bootstraps coreference models by utilizing Wikipedia anchor texts. The basic idea is that two anchor texts pointing to the same Wikipedia page are likely coreferential (See Figure 2 for an example). Our method builds a large distantly-supervised dataset $\mathcal{W}$ for the target language by leveraging this observation:

$$\mathcal{W} = \{D_1, D_2, ..., D_m\}$$

where $D_i$ is a text document constructed from some Wikipedia page written in the target language. The number of mentions in $D_i$ is the same as the number of anchor texts in the text portion of the original Wikipedia article. We consider two mentions in $D_i$ to be coreferential if and only if their corresponding anchor texts point to the same article.

After constructing $\mathcal{W}$, we follow a two-step process similar to the continued training approach. We first train a conference resolution model on $\mathcal{W}$ until convergence. Then, we finetune the pre-trained model on the final target dataset.

Compared to a manually-labeled dataset, $\mathcal{W}$ has several disadvantages. Not all entity mentions are exhaustively marked in Wikipedia documents. For example, in Spanish Wikipedia, pronouns are typically not annotated. Nevertheless, since Wikipedia is one of the largest multilingual repositories for information, $\mathcal{W}$ is generally large (see Table 1 for some statistics), and it contains documents on various topics. As such, $\mathcal{W}$ can still provide some useful distant supervision signals, and so it can serve as a source dataset in the TL process.

### 2.4 Ensemble-Based Coreference Resolution

During the training stage, we train three different coreference resolution models using the TL approaches described above. At test time, we use a simple unweighted averaging method to combine...
the models’ predictions. More specifically, for a candidate span $i$ with no more than $L$ tokens, we compute its mention score as follows:

$$s_{m,\text{ensemble}}(i) = \frac{\left(s_{m,1}(i) + s_{m,2}(i) + s_{m,3}(i)\right)}{3}$$

(9)

where $s_{m,1}(i)$, $s_{m,2}(i)$, and $s_{m,3}(i)$ are the mention scores produced by the three trained models separately (refer to Equation 2). Intuitively, these scores indicate whether span $i$ is an entity mention.

Similar to the process described in Section 2.1, after scoring every span whose length is no more than $L$ using Equation 9, we only keep spans with high mention scores. Then, for each remaining span $i$, we predict a distribution over its antecedents $j \in Y(i)$ as follows:

$$\hat{P}_{\text{ensemble}}(j) = \frac{\exp(s_{\text{ensemble}}(i,j))}{\sum_{k \in Y(i)} \exp(s_{\text{ensemble}}(i,k))}$$

$$s_{\text{ensemble}}(i,j) = \frac{s_1(i,j) + s_2(i,j) + s_3(i,j)}{3}$$

(10)

where $s_1(i,j)$, $s_2(i,j)$, and $s_3(i,j)$ are the pairwise scores produced by the trained models separately (Eq. 3). We fix $s_{\text{ensemble}}(i, e)$ to be 0.

After computing the antecedent distribution for each remaining span, we can extract the final set of mention clusters. Note that while we consider only three individual TL methods in this work, Equation 9 and Equation 10 can easily be extended for the case when we use more TL methods.

### 3 Experiments

#### 3.1 Data and Experiments Setup

**Evaluation metrics** Following prior work (Pradhan et al., 2012), we evaluate coreference using the average $F_1$ between $B^3$ (Bagga and Baldwin, 1998), MUC (Vilain et al., 1995), and CEAF$_{\phi_4}$ (Luo, 2005). We refer to this metric as AVG.

**Datasets** Table 1 shows the basic statistics of all the datasets we used in this work. When using a cross-lingual TL method, we use the English portion of OntoNotes (Pradhan et al., 2012) as the source dataset. We explore three target datasets: OntoNotes Arabic (Pradhan et al., 2012), SemEval Dutch (Recasens et al., 2010), and SemEval Spanish (Recasens et al., 2010). These datasets contain data in three different languages.

#### 3.2 Transformer Encoders

When the target dataset is OntoNotes Arabic, we use GigaBERT (Lan et al., 2020) as the Transformer encoder. GigaBERT is an English-Arabic bilingual language model pre-trained from the English and Arabic Gigaword corpora. When the target dataset is SemEval Dutch or SemEval Spanish, we use the multilingual XLM-RoBERTa (XLM-R) Transformer model (Conneau et al., 2020). More specifically, we use the base version of XLM-R (i.e., xlm-roberta-base).

**Wikipedia-based Dataset Construction** To construct a distantly-supervised dataset, we first download a complete Wikipedia dump in the target language. We then extract clean text and hyperlinks from the dump using WikiExtractor. For each preprocessed article, we cluster its anchor texts based on the destinations of their hyperlinks. We also filter out articles with too few coreference links (e.g., articles that only have singleton mentions).

**General Hyperparameters** We use two different learning rates, one for the lower pretrained Transformer encoder and one for the upper layers. For every setting, the lower learning rate is $1e-5$, the upper learning rate is $1e-4$, and the span length limit $L$ is 30. The number of training/pre-training epochs is set to be 25 in most cases. When pre-training a model on a Wikipedia-based dataset, the number of epochs is 5. When fine-tuning a model already pre-trained on Dutch Wikipedia or Spanish Wikipedia, the number of epochs is 50. During each training/pre-training process, we pick the checkpoint which achieves the best AVG score on the appropriate dev set as the final checkpoint.
Table 2: Overall $F_1$ (in %) on OntoNotes Arabic, SemEval Dutch, and SemEval Spanish.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Prior Work</th>
<th>Approach</th>
<th>Prev. Score</th>
<th>Our Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoNotes Arabic</td>
<td>(Min, 2021)</td>
<td>GigaBERT + C2F + Joint Training</td>
<td>64.55</td>
<td>66.72</td>
</tr>
<tr>
<td>SemEval Dutch</td>
<td>(Xia and Van Durme, 2021)</td>
<td>XLM-R + ICoref + Continued Training</td>
<td>55.40</td>
<td>60.49</td>
</tr>
<tr>
<td>SemEval Spanish</td>
<td>(Xia and Van Durme, 2021)</td>
<td>XLM-R + ICoref + Continued Training</td>
<td>51.30</td>
<td>75.62</td>
</tr>
</tbody>
</table>

Table 3: Test $F_1$ (in %) on the target datasets and the previous SOTA on each dataset (to the best of our knowledge).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Prior Work</th>
<th>Approach</th>
<th>Prev. Score</th>
<th>Our Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dutch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Overall Results

Table 2 shows the overall performance of different approaches. Our baseline approach is to simply train a model with the architecture described in Section 2.1 using only the target dataset of interest. Overall, the performance of a model trained using the baseline approach is positively correlated with the size of the corresponding target dataset, which is expected. A surprising finding is that our baseline approach already outperforms the previous SOTA method for SemEval Spanish (Xia and Van Durme, 2021) by 20.98% in the F1 score. We speculate that the previous SOTA model for SemEval Spanish is severely undertrained.

Table 2 also shows the results of using different TL methods individually. Each of the TL methods can help improve the coreference resolution performance. While continued training seems to be the most effective approach, it requires the existence of a source dataset (OntoNotes English in this case). On the other hand, our newly proposed Wikipedia-based method can help improve the performance without relying on any labeled source dataset.

Finally, Table 2 also shows the results of using different combinations of learning approaches. Our simple unweighted averaging method is effective across almost all model combinations. In particular, by combining all of the three TL methods discussed previously, we can outperform the previous SOTA methods by large margins. In addition, even without using any labeled source dataset, the combination [Baseline Approach ⊕ Wikipedia Pre-Training] can still outperform the previous SOTA.
methods for Arabic and Spanish. This further confirms the usefulness of our Wikipedia-based TL method. Lastly, combining three models trained using the same baseline approach leads to smaller gains than combining the three TL methods. This is expected as ensemble methods typically work best when the individual learners are diverse (Krogh and Vedelsby, 1994; Melville and Mooney, 2003).

### Analysis

#### 3.3.1 How optimal is our simple unweighted averaging method?

Our averaging approach is equivalent to linear interpolation with equal weights. To analyze the optimality of our method, we compare it to the “best possible” interpolation method.

More specifically, we assume that there is an oracle that can tell us which model in an ensemble gives the most accurate prediction for a particular latent variable. Then, for example, suppose we want to score a span \( i \) using an ensemble of three models. If \( i \) is an entity mention, the oracle will tell us that the model that returns the highest mention score for \( i \) is the most accurate. Thus, we can set the score for \( i \) to be \( \max(s_{m,1}(i), s_{m,2}(i), s_{m,3}(i)) \). Following the same logic, if \( i \) is not an entity mention, we will set its score to be \( \min(s_{m,1}(i), s_{m,2}(i), s_{m,3}(i)) \). The same idea can be applied to compute the linking score \( s_{ensemble}(i, j) \) between \( i \) and \( j \).

In Table 2, we see a considerable gap between the performance of our simple averaging method and the oracle-guided interpolation method. Therefore, a promising future direction is to experiment with a more context-dependent ensemble method. Nevertheless, our averaging method is simple, and it does not require any further parameter tuning to combine a set of existing models. Finally, the performance of each oracle-guided ensemble is far from perfect, implying that improving the underlying architecture of each model can also be a worthwhile effort.

### 4 Related Work

#### 4.1 Entity Coreference Resolution

Recently, neural models for entity coreference resolution have shown superior performance over approaches using hand-crafted features. Lee et al. (2017) proposed the first end-to-end neural coreference resolution model named e2e-coref. The model uses a bi-directional LSTM and a head-finding attention mechanism to learn mention representations and calculate mention and antecedent scores. Lee...
et al. (2018) extended the e2e-coref model by introducing a coarse-to-fine pruning mechanism and a higher-order inference mechanism. The model uses ELMo representations (Peters et al., 2018) instead of traditional word embeddings. The model is typically referred to as the c2f-coref model.

Almost all recent studies on entity coreference resolution are influenced by the design of c2f-coref. Joshi et al. (2019) built the c2f-coref system on top of BERT representations (Devlin et al., 2019). Fei et al. (2019) transformed c2f-coref into a policy gradient model that can optimize coreference evaluation metrics directly. Xu and Choi (2020) studied in depth the higher-order inference (HOI) mechanism of c2f-coref. The authors concluded that given a high-performing encoder such as SpanBERT (Joshi et al., 2020), the impact of HOI is negative to marginal. Another line of work aims to simplify and/or reduce the computational complexity of c2f-coref (Xia et al., 2020; Kirstain et al., 2021; Lai et al., 2021; Dobrovolskii, 2021).

The studies mentioned above only trained and evaluated models using English datasets such as OntoNotes English (Pradhan et al., 2012) and the GAP dataset (Webster et al., 2018). On the other hand, there is significantly less work on coreference resolution for other languages. For example, while e2e-coref was introduced in 2017, the first neural coreference resolver for Arabic was only recently proposed in 2020 (Aloraini et al., 2020). For Dutch, many existing systems are still using rule-based (van Cranenburgh, 2019) or traditional learning-based approaches (Hendrickx et al., 2008; De Clercq et al., 2011). Recently, Poot and van Cranenburgh (2020) evaluated the performance of c2f-coref on Dutch datasets of two different domains: literary novels and news/Wikipedia text.

While our models’ architecture is based on e2e-coref (Section 2.1), we go beyond just applying the models to a non-English language in this work. We propose new TL approaches that can take advantage of existing source datasets and Wikipedia to improve the final performance.

### 4.2 Transfer Learning for Coreference Resolution

Compared to English datasets, the size of a coreference resolution dataset for a non-English language is typically smaller. Several recent studies aim to overcome this challenge by applying standard cross-lingual TL methods such as continued training (Kundu et al., 2018; Xia and Van Durme, 2021; Pražák et al., 2021; Min, 2021). These studies only use one transfer method at a time, and they do not explore how to combine multiple TL techniques effectively. Our experimental results (Section 3.2) show that combining various TL techniques can substantially improve the final coreference resolution performance.

A closely related work by Yang et al. (2012) proposed an adaptive ensemble method to adapt coreference resolution across domains. Their study did not explicitly focus on improving coreference resolution for non-English languages. In addition, they experimented with the settings where gold standard mentions are assumed to be provided. We do not make that assumption. Each of our models does both mention extraction and linking.

### 4.3 Leveraging Wikipedia for Coreference Resolution

There have been studies on leveraging Wikipedia for coreference resolution. Eirew et al. (2021) recently created a large-scale cross-document event coreference dataset from English Wikipedia. For cross-document entity coreference, Singh et al. (2012) created Wikilinks by finding hyperlinks to English Wikipedia from a web crawl and using anchor text as mentions. Different from these studies, we focus on within-document entity coreference resolution. In addition, we explore coreference resolution for languages beyond English in this work.

Many previous studies leveraged Wikipedia for related tasks such as name tagging (Alotaibi and Lee, 2012; Nothman et al., 2013; Althobaiti et al., 2014) and entity linking (Pan et al., 2017; Wu et al., 2020a; Cao et al., 2021). We leave the extension of our methods to these tasks for future research.

### 5 Conclusions and Future Work

In this work, we propose an ensemble-based framework that combines various TL techniques. We also introduce a low-cost Wikipedia-based TL approach that does not require any labeled source dataset. Our approaches are highly effective, as our best ensembles achieve new SOTA results for three different languages. An interesting future direction is to explore the use of model compression techniques (Hinton et al., 2015; Han et al., 2016; Lai et al., 2020) to reduce the computational complexity of our ensembles.
6 Limitations

Multilingual language models such as XLM-R (Conneau et al., 2020) and GigaBERT (Lan et al., 2020) are typically pre-trained on large amounts of unlabeled text crawled from the Web. Since these models are optimized to capture the statistical properties of the training data, they tend to pick up on and amplify social stereotypes present in the data (Kurita et al., 2019). Since our coreference resolution models use such pre-trained language models, they may also exhibit social biases present on the Web. Identifying and mitigating social biases in neural models is an active area of research (Zhao et al., 2018; Sheng et al., 2021; Gupta et al., 2022).

In the future, we plan to work on removing social biases from coreference resolution models.

Furthermore, while our proposed methods are highly effective, the performance of our best ensembles is still far from perfect. On OntoNotes Arabic, our best system only achieves an F1 score of 66.72%. Such performance may not be acceptable for some downstream tasks (e.g., information extraction from critical clinical notes).

Finally, even though Wikipedia is available in more than 300 languages, there are still very few Wikipedia pages for some very rare languages. Our proposed methods are likely to be less effective for such rare languages.

References


A Reproducibility Information

In this section, we present the reproducibility information of our paper.

Implementation Dependencies Libraries

Pytorch 1.11.0 (Paszke et al., 2019), Transformers 4.17.0 (Wolf et al., 2020), SentencePiece 0.1.96 (Kudo and Richardson, 2018), PyTorch Metric Learning (Musgrave et al., 2020).

Computing Infrastructure

The experiments were conducted on a server with Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz and NVIDIA Tesla V100 GPUs. GPU memory is 16G.

Number of Model Parameters

When the target dataset is OntoNotes Arabic, we use GigaBERT (Lan et al., 2020) as the Transformer encoder. GigaBERT has about 125M parameters.

When the target dataset is SemEval Dutch or SemEval Spanish, we use the base version of XLM-R (i.e., xlm-roberta-base) (Conneau et al., 2020). xlm-roberta-base has about 278M parameters.

Hyperparameters

The information about the hyperparameters is available in the main paper.
**Expected Validation Performance**  We report the validation performance of the ensemble \([\text{Continued Training} \oplus \text{Joint Training} \oplus \text{Wikipedia Pre-Training}]\).

The validation F1 score of the ensemble for Arabic coreference resolution is 66.60%. The total time needed for the evaluation is about 1 minute and 19 seconds.

The validation F1 score of the ensemble for Dutch coreference resolution is 57.81%. The total time needed for the evaluation is about 20 seconds.

The validation F1 score of the ensemble for Spanish coreference resolution is 75.73%. The total time needed for the evaluation is about 1 minute and 42 seconds.
Contrastive Hierarchical Discourse Graph for Scientific Document Summarization

Haopeng Zhang, Xiao Liu, Jiawei Zhang
IFM Lab, Department of Computer Science, University of California, Davis, CA, USA
haopeng,xiao,jiawei@ifmlab.org

Abstract
The extended structural context has made scientific paper summarization a challenging task. This paper proposes CHANGES, a contrastive hierarchical graph neural network for extractive scientific paper summarization. CHANGES represents a scientific paper with a hierarchical discourse graph and learns effective sentence representations with dedicated designed hierarchical graph information aggregation. We also propose a graph contrastive learning module to learn global theme-aware sentence representations. Extensive experiments on the PubMed and arXiv benchmark datasets prove the effectiveness of CHANGES and the importance of capturing hierarchical structure information in modeling scientific papers.

1 Introduction
Extractive document summarization aims to extract the most salient sentences from the original document and form the summary as an aggregate of these sentences. Compared to abstractive summarization approaches that suffer from hallucination generation problems (Kryściński et al., 2019; Zhang et al., 2022b), summaries generated in an extractive manner are more fluent, faithful, and grammatically accurate, but may lack coherence across sentences. Recent advances in deep neural networks and pre-trained language models (Devlin et al., 2018; Lewis et al., 2019) have led to significant progress in single document summarization (Nallapati et al., 2016a; Narayan et al., 2018; Liu and Lapata, 2019; Zhong et al., 2020). However, these methods mainly focus on short documents like news articles in CNN/DailyMail (Hermann et al., 2015) and New York Times (Sandhaus, 2008), and struggle when dealing with relatively long documents such as scientific papers.

The challenges of lengthy scientific paper summarization lie in several aspects. First, the extended input context hinders cross-sentence relation modeling, the critical step of extractive summarization (Wang et al., 2020). Thus, sequential models like RNN are incapable of capturing the long-distance dependency between sentences, and hard to differentiate salient sentences from others. Furthermore, scientific papers tend to cover diverse topics and contain rich hierarchical discourse structure information. The internal hierarchy structure, like sections, paragraphs, sentences, and words, is too complex for sequential models to capture. Scientific papers generally follow a standard discourse structure of problem definition, methodology, experiments and analysis, and conclusions (Xiao and Carenini, 2019). Moreover, the lengthy input context also makes the widely adopted self-attention Transformer-based models (Vaswani et al., 2017) inapplicable. The input length of a scientific paper can range from 2000 to 7,000 words, which exceeds the input limit of the Transformer due to the quadratic computation complexity of self-attention. Thus, sparse Transformer models like BigBird (Zaheer et al., 2020) and Longformer (Beltagy et al., 2020) are proposed.

Recently, researchers have also turned to graph neural networks (GNN) as an alternative approach. Graph neural networks have been demonstrated to be effective at tasks with rich relational structure and can preserve global structure information (Yao et al., 2019; Xu et al., 2019; Zhang and Zhang, 2020). By representing a document as a graph, GNNs update and learn sentence representations by message passing, and turn extractive summarization into a node classification problem. Among all attempts, one popular way is to construct cross-sentence similarity graphs (Erkan and Radev, 2004; Zheng and Lapata, 2019), which uses sentence representation cosine similarity as edge weights to model cross-sentence semantic relations. Xu et al. (2019) proposed using Rhetorical Structure Theory (RST) trees and coreference mentions to capture cross-sentence discourse relations. Wang et al. (2020) proposed constructing a word-document het-
To address the above challenges, we propose CHANGES (Contrastive Hierarchical Graph neural network for Extractive Summarization), a hierarchical graph neural network model to fully exploit the section structure of scientific papers. CHANGES first constructs a sentence-section hierarchical graph for a scientific paper, and then learns hierarchical sentence representations by dedicated designed information aggregation with iterative intra-section and inter-section message passing. Inspired by recent advances in contrastive learning (Liu and Liu, 2021; Chen et al., 2020), we also propose a graph contrastive learning module to learn global theme-aware sentence representations and provide fine-grained discriminative information. The local sentence and global section representations are then fused for salient sentence prediction. We validate CHANGES with extensive experiments and analyses on two scientific paper summarization datasets. Experimental results demonstrate the effectiveness of our proposed method. Our main contributions are as follows:

- We propose a hierarchical graph-based model for long scientific paper extractive summarization. Our method utilizes the hierarchical discourse of scientific documents and learns effective sentence representations with iterative intra-section and inter-section sentence message passing.

- We propose a plug-and-play graph contrastive module to provide fine-grained discriminative information. The graph contrastive module learns global theme-aware sentence representations by pulling semantically salient neighbors together and pushing apart unimportant sentences. Note that the module could be added to any extractive summarization system.

- We validate our proposed model on two benchmark datasets (arXiv and PubMed), and the experimental results demonstrate its effectiveness over strong baselines.

2 Related Work

2.1 Extractive Summarization on Scientific Papers

Despite the superior performance on news summarization by recent neural network models (Zhou et al., 2018; Zhang et al., 2023a,b; Fonseca et al., 2022) and pre-trained language models (Liu and Lapata, 2019; Lewis et al., 2019), progress in long document summarization such as scientific papers is still limited.

Traditional approaches to summarize scientific articles rely on supervised machine learning algorithms such as LSTM (Collins et al., 2017) with surface features such as sentence position, and section categories. Recently, Xiao and Carenini (2019) proposed a neural-based method by incorporating both the global context of the whole document and the local context within the current topic with an encoder-decoder model. Ju et al. (2021) designed an unsupervised extractive approach to summarize long scientific documents based on the Information Bottleneck principle. Dong et al. (2020) proposed an unsupervised ranking model by incorporating two-level hierarchical graph representation and asymmetrical positional cues to determine sentence importance. Recent works also apply pre-trained sparse language models like Longformer for modeling long documents (Beltagy et al., 2020; Ruan et al., 2022; Cho et al., 2022).

2.2 Graph-based Summarization

Graph models have been widely applied to extractive summarization due to the capability of modeling cross-sentence relations within a document. The sparsity nature of graph structure also brings scalability and flexibility, making it a good fit for long documents. Graph neural networks’ memory costs are generally linear with regard to the input size compared to the quadratic self-attention mechanism.

Researchers have explored supervised graph neural network methods for summarization (Cui and Hu, 2021; Jia et al., 2020; Huang and Kurohashi, 2021; Xie et al., 2022; Phan et al., 2022). Yasunaga et al. (2017) first proposed to use Graph Convolutional Network (GCN) on the approximate discourse graph. Xu et al. (2019) then applied GCN on structural discourse graphs based on RST trees and coreference mentions. Recently, Wang et al. (2020) proposed constructing a word-document heterogeneous graph by using words as the intermediary.
Figure 1: The overall model architecture of CHANGES. We first construct a hierarchical graph for an input document, and then learn representations with a graph contrastive module and hierarchical graph layers. The concatenation representations of sentence node and its section node will be fused for summary sentence selection.

between sentences. Zhang et al. (2022a) proposed to use hypergraph to capture the high-order sentence relations within the document. Our paper follows the series of work but incorporates hierarchical graphs for scientific paper discourse structure modeling and graph contrastive learning for theme-aware sentence representation learning.

3 Method

Given a document $D = \{s_1, s_2, ..., s_n\}$ with $n$ sentences and $m$ sections, we first represent it as a hierarchical graph and formulate extractive summarization as a node labeling task. The objective is to predict labels $y_i \in \{0, 1\}$ for all sentences, where $y_i = 1$ and $y_i = 0$ represent whether the $i$-th sentence should be included in the summary or not, respectively.

The overall model architecture of CHANGES is shown in Figure 1. CHANGES consists of two modules: a graph contrastive learning module to learn global theme-aware sentence representations and a hierarchical graph layer module to learn hierarchical graph node representations with iterative message passing. The learned sentence node and section node representations will be used as indicators for salient sentence selection.

3.1 Graph Construction

Given an academic paper $D$, we first construct a hierarchical graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ stands for the node set and $\mathcal{E}$ represents edges between nodes. In order to utilize the sentence-section hierarchical structure of academic papers, the undirected hierarchical graph $G$ contains both sentence nodes and section nodes, defined by $\mathcal{V} = \mathcal{V}_{\text{sen}} \cup \mathcal{V}_{\text{sec}}$, where each sentence node $v_{\text{sen}, j} \in \mathcal{V}_{\text{sen}}$ represents a sentence $s_j$ in the document $D$ and $v_{\text{sec}, j} \in \mathcal{V}_{\text{sec}}$ represents one section in the document. The edge connection of $G$ is defined as $\mathcal{E} = \mathcal{E}_{\text{sen}} \cup \mathcal{E}_{\text{sec}} \cup \mathcal{E}_{\text{cross}}$, where $\mathcal{E}_{\text{sen}}$ denotes the connection between sentence nodes within the same section, $\mathcal{E}_{\text{sec}}$ denotes the connection between section nodes, and $\mathcal{E}_{\text{cross}}$ denotes the cross-connection between a sentence node and its corresponding section node. Note that we also add a special section supernode $v_D$ that represents the whole document $D$. An illustration of the hierarchical graph is shown in Figure 2.

**Edge Connection** Unlike prior work (Zheng and Lapata, 2019; Dong et al., 2020) that uses cosine similarity of sentence semantic representations as edge weights, we construct unweighted hierarchical graphs to disentangle structural information (adjacency matrix $A$) from semantic information (node representation $H$). In other words, connected nodes have weight 1, and disconnected nodes have...
weight 0 in the adjacency matrix $A$.

Formally, sentence-level edge $e_{\text{sen}i,j}$ connects sentence nodes $v_{\text{sen}i}$ and $v_{\text{sen}j}$ if they are within the same section, aiming to aggregate local intra-sentence information. All section nodes are fully connected by section-level edges $e_{\text{sec}i,p}$, aiming to aggregate global inter-section information. The cross-level edge $e_{\text{cross}i,p}$ connects the sentence node $v_{\text{sen}i}$ to its corresponding section node $v_{\text{sec}i,p}$, which allow message passing between sentence-level and section-level nodes.

In a hierarchical graph, a sentence node could only directly interact with local neighbor nodes within the same section, and indirectly interact with sentence nodes of other sections via section-level node connections.

![Hierarchical Graph](image)

**Figure 2**: An illustration of a hierarchical graph for a long input document with rich discourse structures.

**Node Representation** We adopt BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) as sentence encoder to embed the semantic meanings of sentences $\{s_1, s_2, ..., s_n\}$ as initial node representations $X = \{x_1, x_2, ..., x_n\}$. Note that BERT here is only used for initial sentence embedding, but is not updated during the training process to reduce model computing cost and increase efficiency.

In addition to the semantic representation of sentences, we also inject positional encoding following Transformer (Vaswani et al., 2017) to preserve the sequential order information. We apply the hierarchical position embedding by (Ruan et al., 2022) to model sentence positions accompanying our hierarchical graph. Specifically, the position of each sentence $s_i$ can be represented as two parts: its corresponding section index $p_{i}^{\text{sec}}$, and its sentence index within section $p_{i}^{\text{sen}}$. Formally, the hierarchical position embedding (HPE) of sentence $s_i$ can be calculated as:

$$\text{HPE}(s_i) = \text{PE}(p_{i}^{\text{sec}}) + \text{PE}(p_{i}^{\text{sen}}),$$

where $\text{PE}(\cdot)$ refers to the position encoding function in (Vaswani et al., 2017):

$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d}),$$

$$\text{PE}(pos, 2i + 1) = \cos(pos/10000^{2i/d}).$$

Overall, we can get the initial sentence node representations $H_{\text{sen}}^{0} = \{h_{\text{sen}1}^{0}, h_{\text{sen}2}^{0}, ..., h_{\text{sen}n}^{0}\}$, with vector $h_{i}^{0} \in \mathbb{R}^{d}$ defined as:

$$h_{i}^{0} = x_{i} + \text{HPE}(s_i),$$

where $d$ is the dimension of the node embedding. The initial section node representation $H_{\text{sec}i}^{0} \in \mathbb{R}^{d}$ for the $j$-th section is the mean of its connected sentences embeddings, and the document node representation $H_{\text{doc}}^{0} \in \mathbb{R}^{d}$ is the mean of all section node embeddings.

**3.2 Graph Contrastive Module**

After constructing the hierarchical graph with adjacency matrix $A$ and node representation $H_{\text{sen}}^{0} \in \mathbb{R}^{n \times d}$, we apply a graph contrastive learning (GCL) module to capture global context information. Motivated by the principal idea that a good summary sentence should be more semantically similar to the source document than the unqualified sentences (Radev et al., 2004; Zhong et al., 2020), our GCL module updates sentence representations using Graph Attention Network (Velićković et al., 2017) with a contrastive objective to learn the global theme-aware sentence representations. Note that the module could be added to any extractive summarization system.

**Graph Attention Network** Given a constructed graph $G = (V, E)$ with node representations $H$ and adjacent matrix $A$, a GAT layer updates a node $v_i$ with representation $h_i$ by:

$$e_{ij} = \text{LeakyReLU} (W_a [W_{in} h_i \| W_{in} h_j]),$$

$$\alpha_{ij} = \frac{\exp (e_{ij})}{\sum_{l \in N_i} \exp (e_{il})},$$

$$h_i' = \sigma \left( \sum_{j \in N_i} \alpha_{ij} W_a h_j \right),$$

where $N_i$ denotes the 1-hop neighbors of node $v_i$, $\alpha_{ij}$ denotes the attention weight between nodes $h_i$ and $h_j$, $W_{in}, W_a, W_v$ are trainable weight matrices, and $\|$ denotes concatenation operation.

```
The above single-head graph attention is further extended to multi-head attention, where $T$ independent attention mechanisms are conducted and their outputs are concatenated as:

\[
h'_i = \|_{i=1}^T \sigma \left( \sum_{j \in N_i} \alpha_{ij} W^k_h h_j \right)
\]

\begin{equation}
\end{equation}

**Contrastive Loss** Contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Marelli et al., 2014). Recent works have demonstrated contrastive learning to be effective in high-order representation learning (Chen et al., 2020; Gao et al., 2021). Thus, we optimize our GCL module in a contrastive manner with the following contrastive loss. The goal of contrastive learning is to learn theme-aware sentence embedding by pulling semantically salient neighbors together and pushing apart less salient sentences. The contrastive loss is formally defined as:

\[
L_c = - \log \frac{\exp(sim(h'_D, h'_i)/\tau)}{\sum_{j=1} \exp(sim(h'_D, h'_j)/\tau)}.
\]

\begin{equation}
\end{equation}

where $h'_D$ is the document node embedding, $h'_i$ is the updated representation of sentence $s_i$, and $\tau$ is the temperature factor.

After passing through the GCL module, the learned global theme-aware sentence embeddings $H^c_{sen} = \{h^c_{sen1}, h^c_{sen2}, ..., h^c_{senL}\} \in \mathbb{R}^{n \times d}$ are then passed to the hierarchical graph layer module.

### 3.3 Hierarchical Graph Layer

To exploit the sentence structure of academic papers, CHANGES then updates sentence and section node representations with hierarchical graph layers in an iterative manner.

The hierarchical graph layer first updates sentence embeddings with the local neighbor sentences within the same section with GAT for intra-section message passing, then update section nodes with sentence nodes for cross-level information aggregation to exploit the hierarchical structure of academic papers. Next, inter-section message passing allow global context information interaction. Finally, the sentence nodes are updated based on their corresponding section node, fusing both local and global context information.

Formally, each iteration contains four update processes: one intra-section message passing, one sentence-to-section aggregation, one intersection message passing, and finally one section-to-sentence aggregation. For the $l$-th iteration, the process can be represented as:

\[
H^l_{sen} = \text{GAT}(H_{sen}^l)
\]

\[
H^l_{sec} = \text{GAT}(H^l_{sen})
\]

\[
H^l+1_{?sen} = \sigma(W_h[H^l_{sen} || H^l_{sec}])
\]

\[
H^l+1_{?sec} = \sigma(W_h[H^l_{sen} || H^l_{sec}])
\]

where $H^l_{sen}, H^l_{sec}$ denotes the intermediate representations of sentence and section nodes, $H^l_{sen}, H^l_{sec}$ denotes the updated sentence and section node representations, and $[H^l_{sen} || H^l_{sec}]$ denotes the concatenation of intermediate sentence node representation and its corresponding updated section node representation.

In this way, CHANGES updates and learns hierarchy-aware sentence embeddings through the hierarchical graph layers.

### 3.4 Optimization

After passing $L$ hierarchical graph layers, we obtain the final sentence node representations $H^c_{sen} = \{h^c_{sen1}, h^c_{sen2}, ..., h^c_{senL}\} \in \mathbb{R}^{n \times d}$. We then add a multi-layer perceptron (MLP) followed by a sigmoid activation function to indicate the confidence scores for extracting each sentence in the summary.

Formally, the predicted confidence score $y_i$ to extract a sentence $s_i$ in section $sec_j$ as a summary sentence is:

\[
z_i = \text{LeakyReLU}(W_{o1}[h^L_{seni} || h^L_{secj}]),
\]

\[
y_i = \sigma(W_{o2}z_i),
\]

where $W_{o1}, W_{o2}$ are both trainable parameters, and $[h^L_{seni} || h^L_{secj}]$ denote the concatenation of sentence embedding and its corresponding section embedding. During the inference phase, we will select the $k$ sentences with the highest predicted confidence scores as the extractive summary for the input long document.

<table>
<thead>
<tr>
<th></th>
<th>Arxiv</th>
<th>PubMed</th>
</tr>
</thead>
<tbody>
<tr>
<td># train</td>
<td>201,427</td>
<td>112,291</td>
</tr>
<tr>
<td># validation</td>
<td>6,431</td>
<td>6,402</td>
</tr>
<tr>
<td># test</td>
<td>6,436</td>
<td>6,449</td>
</tr>
<tr>
<td>avg. word/doc</td>
<td>4,938</td>
<td>3,016</td>
</tr>
<tr>
<td>avg. word/summary</td>
<td>203</td>
<td>112,291</td>
</tr>
<tr>
<td>avg. sent/doc</td>
<td>205</td>
<td>220</td>
</tr>
<tr>
<td>avg. sent/summary</td>
<td>5</td>
<td>140</td>
</tr>
</tbody>
</table>

Table 1: Statistics of PubMed and Arxiv datasets.
Since the extractive ground truth labels for long documents are highly imbalanced, we optimize hierarchical graph layers using weighted cross entropy loss following (Xiao and Carennini, 2019) as:

$$
\mathcal{L}_s = -\frac{1}{N_d} \sum_{d=1}^{N} \sum_{i=1}^{N_d} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)),
$$

(11)

where $N$ denotes the number of training instances in the training set, $N_d$ denotes the number of sentences in the document, $\eta = \frac{\#\text{negative}}{\#\text{positive}}$ denote the ratio of the number of negative and positive sentences in the document, and $y_i$ represent the ground-truth of sentence $i$.

### Training Loss

Overall, we optimize CHANGES in an end-to-end manner, by optimizing the graph contrastive module and hierarchical graph layers simultaneously.

The overall training loss of CHANGES is:

$$
\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_c,
$$

(12)

where $\lambda$ is a re-scale hyperparameter and $\mathcal{L}_c$ denotes the contrastive loss in Equation 7.

4 Experiment

4.1 Experiment Setup

**Dataset** To validate the effectiveness of CHANGES, we conduct extensive experiments on two benchmark datasets: arXiv and PubMed (Cohan et al., 2018). The arXiv dataset contains papers in scientific domains, while the PubMed dataset contains scientific papers from the biomedical domain. These two benchmark datasets are widely adopted in long document summarization research and we use the original train, validation, and testing splits as in (Cohan et al., 2018). The detailed statistics of datasets are shown in Table 1.

**Evaluation** Following the common setting, we use ROUGE F-scores (Lin, 2004) as the automatic evaluation metrics. Specifically, we report the ROUGE-1/2 scores to measure summary informativeness and ROUGE-L scores to measure summary fluency. Following prior work (Liu and Lapata, 2019; Nallapati et al., 2016b), we also construct extractive ground truth labels (ORACLE) for training by greedily optimizing the ROUGE score on gold-reference abstracts.

4.2 Implementation Details

We use the publicly released BERT-base 1 (Devlin et al., 2018) as the sentence encoder. The BERT encoder is only used to generate initial sentence embeddings, but is not updated during training to improve model efficiency. We adopt the Graph Attention Network 2 (Veličković et al., 2017) implementation with 8 attention heads and 2 stack layers for graph message passing. The hidden size of our model is set to 2048.

Our model is trained with the Adam optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.0001 and a dropout rate of 0.1. We train our model on a single RTX A6000 GPU for 10 epochs and validate after each epoch using ROUGE-1 F-score. We employ early stopping to select the best model for a patient duration of 3 epochs. We searched the training loss re-scale factor $\lambda$ in the range of 0 to 1 with 0.1 step size and got the best value of 0.5.

4.3 Baseline Methods

We perform a systematic comparison with recent approaches in both extractive and abstractive summarization for completeness. We keep the same train/validation/test splitting in all the experiments and report ROUGE scores from the original papers if available, or scores from (Xiao and Carennini, 2019) otherwise. Specifically, we compare with the following strong baseline approaches:

**Unsupervised methods:** LEAD method that selects the first few sentences as a summary, SumBasic (Vanderwende et al., 2007), graph-based unsupervised models LexRank (Erkan and Radev, 2004), PACSUM (Zheng and Lapata, 2019) and HIPORANK (Dong et al., 2020).

**Neural extractive models:** encoder-decoder based model Cheng&Lapata (Cheng and Lapata, 2016) and SummaRuNNer (Nallapati et al., 2016a); local and global context model ExtSum-LG (Xiao and Carennini, 2019) and its variants ExtSum-LG+RdLoss/MMR (Xiao and Carennini, 2020); language model-based methods SentCLF and SentPTR (Subramanian et al., 2019).

**Neural abstractive models:** pointer network generation model PGN (See et al., 2017), hierarchical attention generation model DiscourseAware (Cohan et al., 2018), and transformer-based generation model TLM-I+E (Subramanian et al., 2019).

---

1. https://github.com/google-research/bert
2. https://github.com/PetarV-/GAT
Table 2: ROUGE F1 results on PubMed and Arxiv datasets. We keep the same train/validation/test splitting in all the experiments and report ROUGE scores from the original papers if available, or scores from (Xiao and Carenini, 2019) otherwise.

4.4 Experiment Results

Table 2 shows the performance comparison of CHANGES and all baseline methods on both PubMed and arXiv datasets. The first blocks include the extractive ground truth ORACLE, position-based sentence selection method LEAD, and other unsupervised baseline approaches. The second block covers state-of-the-art supervised extractive neural baselines, and the third block covers the supervised abstractive baselines.

According to the results, HIPORANK (Dong et al., 2020) achieves state-of-the-art performance for graph-based unsupervised methods. Compared to PACSUM (Zheng and Lapata, 2019), the only difference is that HIPORANK incorporates section structural information for degree centrality calculation. The performance gain demonstrates the significance of capturing the hierarchical structure of academic papers when modeling cross-sentence relations.

Interestingly, the LEAD approach performs far better when summarizing short news like CNN/DailyMail (Hermann et al., 2015) and New York Times (Sandhaus, 2008) than summarizing academic papers, as shown in Table 2. The results show that the distribution of ground truth sentences in academic papers is more even. In other words, academic papers have less positional bias than news articles.

We also notice that the neural extractive models tend to outperform the neural abstractive methods in general, possibly because the extended context is more challenging for generative models during decoding. ExtSum-LG (Xiao and Carenini, 2019) is a benchmarked extractive method with section information by incorporating both the global context of the whole document and the local context within the current topic. We argue that CHANGES could better model the complex sentence structural information with the hierarchical graph than the LSTM-minus in ExtSum-LG.

According to the experimental results, our model CHANGES outperforms all baseline approaches significantly in terms of ROUGE F1 scores on both PubMed and arXiv datasets. The performance improvements demonstrate the usefulness of the global theme-aware representations from the graph contrastive learning module and the hierarchical graph structure for identifying the salient sentences.

5 Analysis

5.1 Ablation Study

We first analyze the influence of different components of CHANGES in Table 3. Here the second row ‘w/o Contra’ means we remove the GCL module and do not update the theme-aware sentence embeddings. The third row ‘w/o Hierarchical’ denotes that we only use the theme-aware sentence embedding for prediction without hierarchical graph layers. As shown in the table, removing either
Table 3: Ablation study results of removing components of CHANGES on PubMed and arXiv datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PubMed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANGES</td>
<td>46.43</td>
<td>21.17</td>
<td>41.58</td>
</tr>
<tr>
<td>w/o GCL</td>
<td>43.91</td>
<td>18.57</td>
<td>40.01</td>
</tr>
<tr>
<td>w/o Hierarchical</td>
<td>43.76</td>
<td>18.30</td>
<td>39.88</td>
</tr>
<tr>
<td>arXiv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANGES</td>
<td>45.61</td>
<td>18.02</td>
<td>40.06</td>
</tr>
<tr>
<td>w/o GCL</td>
<td>44.47</td>
<td>16.58</td>
<td>38.87</td>
</tr>
<tr>
<td>w/o Hierarchical</td>
<td>44.72</td>
<td>16.79</td>
<td>39.10</td>
</tr>
</tbody>
</table>

component causes a significant model performance drop, which indicates that modeling sequential order information, semantic information, and hierarchical structural information are all necessary for academic paper summarization.

Interestingly, the theme-aware sentence embeddings and the hierarchy structure-aware sentence embeddings are almost equally critical to sentence salience modeling. The finding indicates the importance of modeling cross-sentence relations from both semantic and discourse structural perspectives.

5.2 Performance Analysis

We also analyze the sensitivity of CHANGES to section structure and length of academic papers. As shown in Figure 3, we see a performance drop trending when the number of sections increases. This is likely because the complex section structure hinders the inter-section sentence interactions. The model performance on the arXiv dataset is more stable compared to the PubMed dataset although documents in the arXiv dataset are relatively longer. We notice the same trend in Figure 4, model performance is also more stable on arXiv datasets across different document lengths. We argue this may imply that our model is more fit for longer documents that have richer discourse structures.

Regarding the document length, we see a steady performance gain when comparing to benchmark baseline methods ExtSum-LG on both datasets as shown in Figure 4. We also see as the document length increases, the performance gap between CHANGES and extractive summary performance ceiling ORACLE becomes smaller. The finding also verifies that CHANGES is especially effective and fit for long academic papers modeling.
6 Conclusion

In this paper, we propose CHANGES, a contrastive hierarchical graph-based model for scientific paper extractive summarization. CHANGES first learns global theme-aware sentence representations by graph contrastive learning module. Moreover, CHANGES incorporates the sentence-section hierarchical structure by separating intra-section and inter-section message passing and aggregating both global and local information for effective sentence embedding. Automatic evaluation on the PubMed and arXiv benchmark datasets proves the effectiveness of CHANGES and the importance of capturing both semantic and discourse structure information in modeling scientific papers.

In spite of the strong zero-shot performance of large language models like ChatGPT on various downstream tasks, long document modeling is still a challenging problem in the LLM era. Transformer-based GPT-like systems still suffer from the attention computing complexity problem and will benefit from effective and efficient modeling of long documents.

Limitations

In spite of the strong performance of CHANGES, its design still has the following limitations. First, CHANGES only extracts the sentence-section-document hierarchical structure of academic papers. We believe the model performance could be further improved by incorporating document hierarchy of different granularity like dependency parsing trees and Rhetorical structure theory trees. We leave this for future work. In addition, we only focus on single academic paper summarization in this work. Academic papers generally contain a large amount of domain knowledge, thus introducing domain knowledge from peer papers or citation networks should further boost model performance.

Acknowledgment

This work is supported by NSF through grants IIS-1763365 and IIS-2106972. We also thank the anonymous reviewers for their helpful feedback.

References


Leveraging Structural Discourse Information for Event Coreference Resolution in Dutch

Loic De Langhe, Orphée De Clercq, Veronique Hoste
LT3, Language and Translation Technology Team, Ghent University, Belgium
Groot-Britanniëlaan 45, 9000 Ghent, Belgium
firstname.lastname@ugent.be

Abstract
Structural information is known to be important in resolving coreferential relations. We directly embed discourse structure information (subsection, paragraph and text location) in a transformer-based Dutch event coreference resolution model in order to more explicitly provide it with structural information. Results reveal that integrating this type of knowledge leads to a significant improvement in CONLL F1 for within-document settings (+ 8.6%) and a minor improvement for cross-document settings (+ 1.1%).

1 Introduction

Large Language models (LLMs) and transformer-based architectures have significantly changed the domain of Natural Language Processing (NLP) in recent years. Through pre-training and fine-tuning masked language models (MLMs) such as BERT (Devlin et al., 2018), state-of-the-art results can be obtained for tasks requiring deep semantic or syntactic knowledge such as readability assessment (Imperial, 2021), syntactic parsing (He and Choi, 2019) and conversational question-answering (Staliūnaitė and Iacobacci, 2020). However, despite their apparent dominance over other methods, transformer-based language models are still not the ‘one-size-fits-all’ solution for a subset of NLP tasks. In particular, discourse-based tasks such as Event Coreference Resolution (ECR) still pose a major challenge. Within ECR, the goal is to determine whether or not two textual events refer to the same real-life or fictional event, as is the case in Examples 1 and 2.

1. Frankrijk Verslaat België in de halve finales van de FIFA wereldbeker voetbal EN: France beats Belgium in the semi-final of the FIFA world cup.

Understanding that these two Examples, which have been sourced from a Dutch newspaper article, refer in fact to the same real-world occurrence is straightforward for human readers. Tasks like these typically require understanding of long-distance semantic relationships and dependencies within a given text, or even across multiple texts. While human readers can take advantage of both their extensive extra-linguistic knowledge and structural awareness of the text, AI algorithms typically do not possess such skills. For transformer-based language models in particular long-distance semantic dependencies throughout texts might pose a particular problem. Because MLM pre-training is typically limited to the immediate (sentence) context, the model is unable to learn these dependencies. Additionally, while models such as ALBERT (Lan et al., 2019) have tried to explicitly integrate textual and discourse structure in transformer-based architectures, these models still tend to focus on immediate local context and not on the discourse as a whole.

These limitations pose significant problems for ECR. Recent work has indicated that the correct classification of coreferential links between events in BERT-based models is primarily dependent on the outward lexical similarity of those events (De Langhe et al., 2023b). While logical in principle, this is highly problematic in cases such as Example 3 and Example 4, as there exist many instances in which two events are lexically similar, but that do not corefer.

3. De Franse president Macron ontmoette de Amerikaanse president voor de eerste keer vandaag EN: The French president Macron met with the American president for the first time today
4. Frans President Sarkozy ontmoette de Amerikaanse president EN: French President Sarkozy met de American president
Vice versa, non-similar event mentions are not necessarily not in a coreferential relation. Although the latter cases are more exceptional, the overall sparseness of ECR results makes that the bulk of training data often consists of similar, but not-coreferring event mentions. Overall, transformer-based approaches have made significant strides within the field of ECR (Joshi et al., 2020). Nonetheless, the over-reliance on lexical similarity between event mentions might impede further improvement. Interestingly, earlier feature-based studies have shown that integrating certain structural features, such as the proximity of two events in a given text can have a positive effect (Lu and Ng, 2018). We aim to improve an existing Dutch transformer-based ECR model (De Langhe et al., 2022b) by enriching it with structural discourse-level information. The goal of this paper is two-fold. First, it is our ambition to illustrate that concepts rooted in general linguistic theory and fundamental to our own understanding of coreferential relations can also improve the performance of LLMs on this task. Second, we wish to address the gap between ECR studies in the English language domain and those in lower-resourced languages. Currently, there exists very little data or available research for languages other than English. In our experiments we show that including discourse-level information leads to a significant and consistent improvement for within-document ECR models. We also note minor improvements in cross-document contexts.

2 Related Work

There exist two important model paradigms within the domain of ECR. First, mention-ranking approaches focus on finding all possible antecedents for a given event and on generating a ranking of those antecedents based on the likelihood of coreference with the event in question. In Lu and Ng (2017a) a feature-based probabilistic model was used for within-document ECR. The authors show that lexical features such as full or partial overlap of events and cosine similarity between event mentions are among the most important information sources of the model. In addition, they revealed that distance-based features such as the number of sentences between two events also have a noticeable positive effect on the classifier’s performance. A second and more important series of models are mention-pair approaches. This method generates all possible event pairs and reduces the classification to a binary decision (coreferring or not-coreferring) between each event pair. Earlier models within this paradigm were entirely feature-based and relied on a series of lexical, structural and logical constraining features. A large variety of classical machine learning algorithms has been tested using the mention-pair paradigm such as decision trees (Cybulska and Vossen, 2015), support vector machines (Chen et al., 2015) and standard deep neural networks (Nguyen et al., 2016). More recent work has focused on the use of LLMs and transformer encoders (Cattan et al., 2021a,b), with span-based architectures attaining the best overall results (Joshi et al., 2020; Lu and Ng, 2021). It has to be noted that mention-pair approaches relying on LLMs suffer most from the limitations discussed in Section 1. Therefore, recent studies have attempted, with some success, to integrate insights regarding discourse coherence (Held et al., 2021) and domain-specific document discourse information (Choubey et al., 2020) into existing pipelines. Research for comparatively lower-resourced languages has generally followed the paradigms and methods described above and has focused on languages such as Chinese (Mitamura et al., 2015), Arabic (NIST, 2005) and Dutch (Minard et al., 2016).

3 Experimental setup

3.1 Data

Our data consists of a subset of the Dutch ENCORE corpus (De Langhe et al., 2022a), which in its totality consists of 15,407 annotated events spread over 1,015 documents that were sourced from a collection of Dutch (Flemish) newspaper articles. Coreferential relations between events were annotated at both the within- and cross-document level. For the research presented in this paper a
subset of 8,794 events was selected which all come from documents for which a detailed discourse structure was available. Each of these documents can be broken down into subsections, which in turn consist of a number of paragraphs. Subsections are preceded by a subtitle in bold and typically group together a piece of related information. Figure 1 visualizes the general structure of a document in the ENCORE corpus.

For each event in our dataset we thus know at which subsection and paragraph level it is located within a given article. Additionally, for each event we can derive its Text Location, depending on where in the article the event is located. There are 5 possible locations, being the Article Header, Article Subheader, Article Introduction, Subsection Title and Paragraph.

3.2 Experiments

In our experiments we draw inspiration from earlier work on feature-based models (Lu and Ng, 2018) and integrate specific event proximity and structural information into a state-of-the-art Dutch transformer-based ECR model (De Langhe et al., 2023b). We focus on the usage of the readily available discourse and document-level information which was described in Section 3.1.

3.2.1 Baseline coreference algorithm

The ECR model consists of the fine-tuned Dutch BERT model BERTje (de Vries et al., 2019). For this model, the mention-pair approach has demonstratively better results compared to other existing methods (Lu and Ng, 2018, 2021). Concretely, pairwise scores for each pair of event mentions in the dataset are obtained. First, each possible event pair in the data is encoded by concatenating and tokenizing the two events and by subsequently feeding these to the BERTje encoder. A special [SEP] token is inserted between the two event mentions to indicate where one event ends and the other begins. We use the token representation of the classification token [CLS] as the aggregate embedding of each event pair, which is subsequently passed to a softmax-activated classification function. Finally, the results of the binary text pair classification are passed through a clustering algorithm in order to obtain output in the form of coreference chains.

3.2.2 Discourse Embeddings Model

In our proposed algorithm discourse-level positional information (paragraph, subsection and text location) is passed to BERT’s first encoder layer for each individual event during the fine-tuning process. This is done in a similar way as how the positional, segment and token embeddings are used in the original BERT implementation. We believe that this structural information corresponds well with established general theories on discourse structure where related concepts are usually found within close proximity of each other, be it at the sentence, paragraph or section level (Hoeken and Van Vliet, 2000; Glasbey, 1994). By directly integrating this knowledge into the model it would ideally learn that, overall, coreferring mentions are grouped closer together compared to non-coreferring mentions at the document level. As mentioned before in Section 2, it has already been shown that knowledge regarding the proximity of two events can have a positive impact on the classification decision in feature-based models (Lu and Ng, 2017b, 2018). Earlier research has also shown that currently this knowledge is not encoded by BERT-like models (De Langhe et al., 2023a). These findings led us to believe that this specific knowledge can be leveraged by the model to learn about a fundamental aspect of coreferential relations, as well as
Table 1: Subtables report average CONLL results and standard deviation over 3 trials using different random seeds for the discourse-level embeddings in a within-document and cross-document setting respectively. All results in the cross-document table, except the baseline model, automatically include document-level embeddings.

<table>
<thead>
<tr>
<th>Model</th>
<th>CONLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERTjeBaseline</td>
<td>0.432</td>
</tr>
<tr>
<td>BERTjeParagraph</td>
<td>0.466 ± 0.012</td>
</tr>
<tr>
<td>BERTjeSubsection</td>
<td>0.517 ± 0.008</td>
</tr>
<tr>
<td>BERTjeText Location</td>
<td>0.424 ± 0.032</td>
</tr>
<tr>
<td>BERTjeParagraph + Subsection</td>
<td>0.518 ± 0.009</td>
</tr>
<tr>
<td>BERTjeParagraph + TextLocation</td>
<td>0.434 ± 0.028</td>
</tr>
<tr>
<td>BERTjeSubsection + TextLocation</td>
<td>0.437 ± 0.019</td>
</tr>
<tr>
<td>BERTjeParagraph + Subsection + Text Location</td>
<td>0.516 ± 0.022</td>
</tr>
</tbody>
</table>

(a) Results for the Within-document setting

<table>
<thead>
<tr>
<th>Model</th>
<th>CONLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERTjeBaseline</td>
<td>0.519</td>
</tr>
<tr>
<td>BERTjeParagraph</td>
<td>0.530 ± 0.014</td>
</tr>
<tr>
<td>BERTjeSubsection</td>
<td>0.517 ± 0.011</td>
</tr>
<tr>
<td>BERTjeText Location</td>
<td>0.460 ± 0.009</td>
</tr>
<tr>
<td>BERTjeParagraph + Subsection</td>
<td>0.481 ± 0.032</td>
</tr>
<tr>
<td>BERTjeParagraph + TextLocation</td>
<td>0.476 ± 0.048</td>
</tr>
<tr>
<td>BERTjeSubsection + TextLocation</td>
<td>0.468 ± 0.064</td>
</tr>
<tr>
<td>BERTjeParagraph + Subsection + Text Location</td>
<td>0.472 ± 0.026</td>
</tr>
</tbody>
</table>

(b) Results for the Cross-document setting

break away from its aforementioned dependency on outward lexical similarity of events.

In our implementation, all possible subsection, paragraph and text location levels are encoded using a tokenizer-like mechanism where each level of the respective subsection, paragraph or text location is assigned a unique ID, much like individual tokens are encoded using BERT’s own tokenizer. Then, an input embedding matrix of size \( A \times 768 \) is randomly initiated for each type of segment information (subsection, paragraph and text location), where \( A \) is the maximum depth level of a given segment across the dataset and 768 is the standard embedding length for a BERT Baseline model. Concretely, the maximum depth at the paragraph level is 10 if the longest document across the dataset (in number of paragraphs) has 10 paragraphs. The resulting input embedding matrix will then be of dimension \( 10 \times 768 \) and a total of 7680 trainable parameters (\( A \times 768 \)) will be added to the model. The first paragraph in each document is encoded by the same unique ID (i.e., 1) and the paragraph-level embedding for each individual token is obtained by embedding the unique ID through the generated input embedding matrix. The same process is followed for the subsection and text location embeddings. Finally, the resulting discourse-level embeddings are summed with the token, segment and positional embeddings to obtain the input for the first encoder layer. As is the case in the original BERT implementation, the weights of our custom discourse input embedding matrices are also optimized during the fine-tuning process. A high-level visualization of the integration of a paragraph embedding can be found in Figure 2. Subsection and text location embeddings are implemented in an analogous manner.

While, intuitively, our proposed structural embeddings would most likely be most useful in a within-document setting, we also include results for a cross-document setting in order to gauge the effectiveness of discourse-level features in those contexts specifically. Our setup for cross-document ECR is identical to the one described above with the notable exception that we add a fourth type of discourse-level embedding, namely a document embedding. When events are found within the same document this embedding is identical.

4 Results and discussion

Tables 1a and 1b show the results of testing various discourse-level embeddings in a within-document and cross-document context, respectively. We evaluate our results using the established CONLL metric, which is an average of 3 commonly used metrics for coreference evaluation: MUC (Vilain et al., 1995), B\(^3\) (Bagga and Baldwin, 1998) and CEAF (Luo, 2005). We report the average and standard deviations of 3 runs of experiments with different random seeds for the discourse-level input embedding matrices. For the within-document experiments, we see a significant impact on overall performance when including paragraph- and subsection-level information in the fine-tuning process. A combination of paragraph embeddings and subsection embeddings provides the best results. Conversely, we find that the inclusion of Text Location embeddings does not have any noticeable impact on the classification of within-document event coreference.

In the cross-document setting, we find that including structural discourse information does not have a significant impact on classifier performance. While including document and paragraph-level embeddings results in a minor improvement over the baseline coreference model, we find that in general
including discourse-specific embeddings does not help with cross-document event coreference.

5 Conclusion and Future Work

In this paper we explored the potential of using discourse-level embeddings in transformer-based models for event coreference resolution. Motivated by general linguistic theory on the overall structure of language we integrate paragraph, subsection and text location information in a Dutch BERT-based mention-pair event coreference algorithm. We find that in within-document contexts the inclusion of discourse-level information has a significant positive effect on overall classifier performance. In particular, the inclusion of paragraph and subsection information consistently leads to better results. Results for the cross-document setting show only minimal improvement over the baseline model. In future work, we aim to further develop structurally informed models for event coreference resolution as well as look into improving the existing cross-document setup.

Acknowledgements

This work was supported by Ghent University under grant BOFGOA2018000601 and by the Research Foundation–Flanders under project grant number FWO.OPR.2020.0014.01.

References


Loic De Langhe, Orphée De Clercq, and Veronique Hoste. 2023a. What does bert actually learn about event coreference? probing structural information in a fine-tuned dutch language model. Accepted.

Loic De Langhe, Orphée De Clercq, and Veronique Hoste. 2022b. Investigating cross-document event coreference for dutch.


Entity Coreference and Co-occurrence Aware Argument Mining from Biomedical Literature

Boyang Liu\textsuperscript{1}, Viktor Schlegel\textsuperscript{1,2}, Riza Batista-Navarro\textsuperscript{1}, Sophia Ananiadou\textsuperscript{1}
\textsuperscript{1} Department of Computer Science, The University of Manchester, UK
\textsuperscript{2} ASUS Intelligent Cloud Services (AICS), Singapore
{boyang.liu-2@postgrad., riza.batista@, sophia.ananiadou@}manchester.ac.uk
viktor_schlegel@asus.com

Abstract

Biomedical argument mining (BAM) aims at automatically identifying the argumentative structure in biomedical texts. However, identifying and classifying argumentative relations (AR) between argumentative components (AC) is challenging since it not only needs to understand the semantics of ACs but also need to capture the interactions between them. We argue that entities can serve as bridges that connect different ACs since entities and their mentions convey significant semantic information in biomedical argumentation. For example, it is common that related AC pairs share a common entity. Capturing such entity information can be beneficial for the Relation Identification (RI) task. In order to incorporate this entity information into BAM, we propose an Entity Coreference and Co-occurrence aware Argument Mining (ECCAM) framework based on an edge-oriented graph model for BAM. We evaluate our model on a benchmark dataset and from the experimental results we find that our method improves upon state-of-the-art methods.

1 Introduction

There is a growing interest in evidence-based decision making in the biomedical field, as it can assist medical practitioners in selecting the best treatment for a given medical case. However, extracting relevant evidence from vast amounts of biomedical publications is time-consuming for practitioners. Thus, biomedical Argument Mining (BAM), which is the application of Argument Mining (AM) to biomedical texts, is proposed to automatically extract argumentative structures in biomedical texts by identifying Argument Components (AC) and Argument Relations (AR) between ACs (Mayer et al., 2020). BAM includes three primary tasks (Si et al., 2022): (1) argument component identification (ACI)—i.e., distinguishing argumentative components from non-argumentative content; (2) argument component classification (ACC)—categorizing ACs into different types (e.g., claim, and evidence); and (3) relation identification (RI)—recognizing ARs (e.g., support, attack, or none) between a pair of ACs.

Among these tasks, the RI task is the hardest one and existing models tend to underperform on this task, compared to the ACI and ACC tasks (Mayer et al., 2020; Galassi et al., 2021; Si et al., 2022). One possible reason is that these models do not incorporate the information about the co-occurrence of common entities between different ACs. This is a valuable source of semantic information and can be particularly important in BAM. As shown in Fig 1, the AC pairs connected by an AR share entities with coreference relations. Furthermore, entity co-occurrence suggests the direction of the ARs (i.e. ACs with several entities are usually the tail of ARs, like in Figure 1).

Based on this intuition, we propose an Entity Coreference and Co-occurrence aware Argument Mining (ECCAM) framework that effectively captures ARs through entity coreference and co-occurrence. ECCAM is a graph-based model. We build a heterogeneous graph that consists of nodes that represent ACs and entities, and edges between nodes. The entity nodes can serve as bridges that connect different ACs. Considering that the entity coreference and co-occurrence relations exist between nodes and thus are represented by edge embeddings, we employ an edge-oriented graph model (Christopoulou et al., 2019) that learns edge representations of any two connected nodes by combing all paths between the two nodes. This enables information flow between different relations and iteratively updates the edge representations, which is finally used as the representations of ARs. Here, the edges between AC nodes and entity nodes are used to pass the entity coreference information while the edges between entity nodes aim to leverage entity co-occurrence information.
2. The quality of life improved in control group by 13.7%, while combined treatment group showed improvement of 83.5% (p < 0.05 vs. control group).

3. Cumulative survival rate was also significantly higher in the combined treatment group.

4. Intraperitoneal perfusion of verapamil enhances the efficacy of chemotherapy drugs, prolongs survival, and improves the quality of life.

Figure 1: Part of the argument structure of the abstracts from PubMed 23589316. Each text in the rectangle represents an AC. Two ACs connected with an arrow means that there is an AR between them. Entities with the same colour are in the same coreference clusters.

Our contributions are shown below:

- To our best knowledge, this paper is the first to incorporate entity coreference and co-occurrence information into an argument mining model.
- We propose the ECCAM framework based on an edge-oriented model to leverage the entity coreference and co-occurrence information.
- Experimental results show that the entity coreference and co-occurrence information can improve the performance of the RI task significantly.

2 Related Work

Recently, the research community has shown growing interest in the task of BAM. Mayer et al. (2018) created a dataset by annotating ACs within randomized controlled trial abstracts and employ the Sub-Tree Kernel to classify the types of ACs with Bag-Of-Words of biomedical text as input. Further, a dataset with both ACs and ARs are created by Mayer et al. (2020) to deal with three tasks of BAM: ACI, ACC and RI. Various contextualized word embeddings, such as BERT (Devlin et al., 2019), BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019a), and RoBERTa (Liu et al., 2019) are explored to address these tasks. Liu et al. (2022) incorporate zoning information (such as background, result and conclusion) to tackle the ACI and ACC tasks at the same time. Galassi et al. (2021) employed a multi-task framework with an attentive residual network to address the ACC task, RI task, and link prediction task of BAM, based on the assumption that ACs had been detected. SeqMT (Si et al., 2022) also assumes the ACI task is solved and pays attention to the ACC and RI tasks. It utilises a multi-task learning approach to benefit from the sequential dependency between the ACC and RI tasks by transferring the representation of the input and output of the ACC task to the RI task.

However, none of the previous models leverages the entity coreference and co-occurrence information for the RI task, which is the focus of this paper.

3 Model Architecture

Following previous models (Galassi et al., 2018; Si et al., 2022; Galassi et al., 2021), we assume that the outputs of the ACI task are provided, i.e., all ACs have been detected without AC types. Inspired by Christopoulou et al. (2019) who employ an edge-oriented graph model (Christopoulou et al., 2018) to leverage interactions among sentences that share the same entities for document-level relation extraction, we propose a framework for the RI task based on a similar edge-oriented model. Our framework contains three parts: the entity cluster extraction module, the document encoder module and the entity co-occurrence and coreference-aware argument mining module.

3.1 Entity Cluster Extraction Module

Since we cannot assume golden annotation of entities and their mentions, the first step of our framework is to identify all named entities in the AM dataset. One simple way is to train a model on biomedical coreference resolution datasets and use such model to predict entity clusters. However, most biomedical coreference resolution datasets concern diseases (Doğan et al., 2014; Li et al., 2015), species (Gerner et al., 2010; Paflis et al., 2013) or proteins and genes (Wei et al., 2015; Collier and Kim, 2004). Most notably, datasets with entity annotations related to cancer experiments, such as “quality of life” and “survival rate”, are absent from the literature, while the biomedical argument mining dataset (Mayer et al., 2020) is about the cancer research. Thus, we choose another method that first predicts the entities in the AM datasets, and then disambiguates them by mapping to the Unified Medical Language System (UMLS) (Bodenreider, 2004) to obtain a unique identifier of a medical concept to obtain entity clusters.

Specifically, we use the Transformed NER model (Stylianou and Vlahavas, 2021) to obtain entities in the abstracts. This model is trained on the EMB-NLP (Nye et al., 2018) dataset and thus
extracts four types of entities, namely Patient, Intervention, Comparison and Outcome (PICO). Since ACs mainly exist in sentences that describe experimental results and conclusions (Liu et al., 2022), we only use outcome entities, to avoid noise. Then, we use all the extracted entities as input of the QuickUMLS tool (Soldaini and Goharian, 2016) to obtain the corresponding UMLS identifiers (IDs). It is worth mentioning that given one mention, the QuickUMLS usually returns multiple IDs. We handle this situation as follows: when there is an exact match between the predicted entities and the given mention, we will use the ID of the exactly matched entity as the ID for the mention. If there is no exact match entity, we will reserve all the entities whose Jaccard similarity score with the given mention is higher than a specific threshold.

All entities in a document that share the same UMLS ID in a document form an entity cluster. The extracted entity clusters are denoted as $C = \{C_1, C_2, ..., C_l\}$, where $C_i = \{m_1, m_2, ..., m_l\}$, and $l$ is total number of mentions in the $i$-th cluster.

### 3.2 Document Encoder Module

Given a document $D = \{t_1, t_2, ..., t_n\}$ consisting of $n$ tokens as the input of our framework, a SciBERT model is employed as the encoder to generate the embeddings of each mention $m_i$ by averaging all token embeddings $\{x_1, x_2, ..., x_M\}$ in $m_i$. Similarly, the embeddings of each AC $n_a$ and each entity $n_e$ are also an average of the tokens in the AC and entity, respectively.

### 3.3 Graph Construction

We initially construct a heterogeneous graph that consists of two different types of nodes (AC nodes and entity nodes) and three types of edges between the nodes. The rules for edge generation are outlined below.

**AC-AC edge.** If two ACs are adjacent, an edge will connect the two AC nodes. There are two situations where two ACs are adjacent. The first one is that the sequences of this two ACs are adjacent in the document. Another situation is that the words between the two ACs in the document are all non-argumentative. This type of edge is used to learn the context.

**AC-Entity edge.** If an AC mentions an entity at least once, there will be an edge between the entity and the related AC. This type of edge is used to learn the coreference information.

**Entity-Entity edge.** We connect all entity pairs so that the model can learn which co-occurrence of entity pairs is helpful for the RI task.

We use a concatenation operation to get the representation of an edge $e_{ij} = [n_i, n_j]$ given the representations of the source node $n_i$ and the destination node $n_j$ of the edge, where $n_i, n_j \in n_e \cup n_a$.

### 3.4 Entity Co-occurrence and Coreference aware Argument Mining model Module

Given the constructed graph, we employ an edge-oriented graph model (Christopoulou et al., 2018) to leverage the entity coreference and co-occurrence information. The model uses a two step method to iteratively update the edge embeddings of two nodes based on the paths between the two nodes.

First, a path between two nodes $i$ and $j$ is generated using intermediate nodes $k$. Then, the representations of two consecutive edges $e_{ik}$ and $e_{kj}$ are combined by a modified bilinear transformation. Through this action, an edge representation of double the length is generated. All existing paths between $i$ and $j$ through $k$ are combined. The $i$, $j$, and $k$ nodes can be either entity nodes or AC nodes. Intermediate nodes without adjacent edges to the target nodes are ignored. Formally, this is written as:

$$f(e_{ik}^{(l)} \cdot e_{kj}^{(l)}) = \sigma(e_{ik}^{(l)} \otimes W e_{kj}^{(l)})$$

where $\sigma$ is the sigmoid non-linear function, $W$ is a learned parameter matrix, $\otimes$ represents element-wise multiplication, $l$ denotes the length of the edge and $e_{ik}$ refers to the representation of the edge between nodes $i$ and $k$.

At the second step, the original (short) edge representation and the new (longer) edge representation resulting from Equation 2 is aggregated as follows:

$$e_{ij}^{(2l)} = \beta e_{ij}^{(l)} + (1 - \beta) \sum_{k \neq i,j} e_{ik}^{(l)} \cdot e_{kj}^{(l)}$$

56
where $\beta \in [0, 1]$ is a scalar used to assign the weight of the shorter edge representation.

A finite number $N$ of iterations is conducted for the two steps. The final length of path is directly proportional to the number of iterations. After $N$ iterations, the number of edges of the longest path will be up to $2^N$.

### 3.5 Classification Module

Finally, to classify relations between AC pairs, we incorporate a softmax classifier which takes the AC-to-AC edges $e_{aa}$ as input:

$$y = \text{softmax}(W e_{aa} + b)$$

(4)

where $W$ and $b$ are learned parameters of the classification layer. The whole model is trained end-to-end by minimising the cross-entropy loss between predicted and gold ACs.

### 4 Experiment

#### 4.1 Datasets

Following Si et al. (2022), we use the AbstRCT (Mayer et al., 2020) benchmark to evaluate our model and compare it with previous approaches. The AbstRCT dataset is composed of three categories of ACs (major claim, claim, and evidence) and two kinds of ARs (support and attack). It consists of three parts, with the largest being the neoplasm corpus, which is divided into training, development, and testing sets. Moreover, there are two additional test sets. The first one solely consists of abstracts related to glaucoma, while the second one is a mixed set containing 20 abstracts for each disease in the dataset (neoplasm, glaucoma, hypertension, hepatitis, and diabetes).

<table>
<thead>
<tr>
<th>Documents</th>
<th>All ARs</th>
<th>Avg. AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neo_train</td>
<td>350</td>
<td>1427</td>
</tr>
<tr>
<td>Neo_dev</td>
<td>50</td>
<td>210</td>
</tr>
<tr>
<td>Neo_test</td>
<td>100</td>
<td>424</td>
</tr>
<tr>
<td>Gla_test</td>
<td>100</td>
<td>367</td>
</tr>
<tr>
<td>Mix_test</td>
<td>100</td>
<td>329</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the AbstRCT dataset. The data statistics of the three test sets are reported separately. Here, Neo, Gla and Mix represent neoplasm, glaucoma and mixed, respectively.

#### 4.2 Implementation

We use the same train-development-test split for the AbstRCT dataset as was used in Si et al. (2022). We fine-tune cased SciBERT (Beltagy et al., 2019b) and set the maximum sequence length to 256. A learning rate of $2 \cdot 10^{-5}$ is used. We train for 50 epochs with early stopping to avoid overfitting. Our model is implemented in PyTorch (Paszke et al., 2019). We employ an AdamW optimizer (Loshchilov and Hutter, 2019) for parameter optimization and report the macro-averaged F1 scores of models trained with three different random seeds.

#### 4.3 Baselines

In order to evaluate our proposed method, we compare it with the following baselines:

- **ResArg** (Galassi et al., 2018) is a hybrid of residual networks and long short-term memory network. This model is designed to tackle both the ACC and RI tasks simultaneously.

- **ResAttArg** (Galassi et al., 2021) is an upgraded version of ResArg model featuring an attention module. ResArg and ResAttArg have two versions: an average version that calculates the final scores as an average of scores from 10 distinct networks trained with 10 different seeds, and an ensemble version that assigns the class based on the majority vote of the same 10 networks.

- **SeqMT** (Si et al., 2022) utilises a multi-task learning approach to benefit from the sequential dependency between the ACC and RI tasks. It transfers the representation of the input and output of the ACC task to the RI task.

#### 4.4 Main Results

We report the main results in Table 2. It can be observed that our model improves upon the state-of-the-art on two of three test sets even though our model is a single task model while all other baselines are multi-task/ensemble models. To be specific, our model outperforms the current best model on the neoplasm test set by 1.68% F1 score and the mixed test set by 1.11% F1 score. However, there is a gap between the performance of our model and SeqMT on the glaucoma test set. This might be due to the lack of multi-task training: compared with the results of SeqMT, the performance of the single task version model of SeqMT(SeqMT(-Lacc)) similarly experiences a large drop of performance of 8.44% F1 score. Without the additional signal
Table 2: Main results of different models. The best scores are marked in bold. All the results of baselines are copied from the related papers. Here, NEO, GLA and MIX represent neoplasm, glaucoma and mixed.

<table>
<thead>
<tr>
<th>models</th>
<th>NEO</th>
<th>GLA</th>
<th>MIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA(avg)</td>
<td>59.15</td>
<td>57.23</td>
<td>60.31</td>
</tr>
<tr>
<td>RA(Ensemble)</td>
<td>63.16</td>
<td>61.86</td>
<td>63.47</td>
</tr>
<tr>
<td>RAA(avg)</td>
<td>66.49</td>
<td>62.68</td>
<td>63.47</td>
</tr>
<tr>
<td>RAA(Ensemble)</td>
<td>70.92</td>
<td>68.40</td>
<td>67.66</td>
</tr>
<tr>
<td>SeqMT(-\mathcal{L}_{acc})</td>
<td>68.58</td>
<td>64.83</td>
<td>70.30</td>
</tr>
<tr>
<td>SeqMT</td>
<td>71.24</td>
<td>73.27</td>
<td>72.71</td>
</tr>
<tr>
<td>ECCAM</td>
<td>72.92</td>
<td>68.96</td>
<td>73.82</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of our model. ECCAM(-EE) is a model where the edges between the entities are excluded. ECCAM(-EA) removes both edges between entities and between entities and ACs. The best scores are marked in bold.

<table>
<thead>
<tr>
<th>iterations</th>
<th>DEV</th>
<th>NEO</th>
<th>GLA</th>
<th>MIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = 1</td>
<td>60.20</td>
<td>65.32</td>
<td>53.49</td>
<td>66.90</td>
</tr>
<tr>
<td>N = 2</td>
<td>62.75</td>
<td>71.34</td>
<td>58.67</td>
<td>71.14</td>
</tr>
<tr>
<td>N = 3</td>
<td>66.04</td>
<td>71.84</td>
<td>63.37</td>
<td>71.80</td>
</tr>
<tr>
<td>N = 4</td>
<td>67.31</td>
<td>72.92</td>
<td>68.96</td>
<td>73.82</td>
</tr>
<tr>
<td>N = 5</td>
<td>69.64</td>
<td>73.85</td>
<td>66.30</td>
<td>68.83</td>
</tr>
</tbody>
</table>

4.5 Ablation Study

To validate the effects of the entity coreference and entity co-occurrence information, we conduct two ablation experiments. ECCAM(-EE) is a model where the edges between the entities are excluded to test whether the entity co-occurrence information can improve the performance of RI. ECCAM(-EA) aims to reveal the impact of entity coreference information by removing both edges between entities and between entities and ACs. The results in Table 3 show the effectiveness of the entity coreference and entity co-occurrence information. Without the entity co-occurrence information, the performance of our model drops by 0.84%, 2.52% and 1.43% F1 score on the neoplasm, glaucoma and mixed test sets, respectively. The performance of ECCAM(-EA) decreases even more significantly—2.3%, 5.12% and 4.09% F1 score on the neoplasm, glaucoma and mixed test sets—showing the positive impact of entity coreference information.

5 Conclusion

In this paper, we propose the ECCAM model that is based on an edge-oriented graph model (Christopoulou et al., 2019) to incorporate entity coreference and co-occurrence information into BAM. We introduce edges between entity nodes and AC nodes in a heterogeneous graph to help our model capture entity coreference and co-occurrence information respectively. Experiments on the AbstRCT dataset show the effectiveness of these two types of information for the RI task. In the future, we will apply our method to other argument mining domains, such as student essays (Eger et al., 2017).

Limitations

Although our model improves upon state-of-the-art methods of BAM by incorporating entity coreference and co-occurrence information, there are still some limitations to our model. First, it is not
easy to apply our model to other domains where no coreference resolution tool is available. Second, the number of nodes and edges of the generated heterogeneous graph will become enormous if the documents are long and many entities are extracted, which requires more GPU resources.

References


A Weakly-Supervised Learning Approach to the Identification of "Alternative Lexicalizations" in Shallow Discourse Parsing

René Knaebel
Applied Computational Linguistics
Department of Linguistics
University of Potsdam
Germany
rene.knaebel@uni-potsdam.de

Abstract

Recently, the identification of free connective phrases as signals for discourse relations has received new attention with the introduction of statistical models for their automatic extraction. The limited amount of annotations makes it still challenging to develop well-performing models. In our work, we want to overcome this limitation with semi-supervised learning from unlabeled news texts. We implement a self-supervised sequence labeling approach and filter its predictions by a second model trained to disambiguate signal candidates. With our novel model design, we report state-of-the-art results and in addition, achieve an average improvement of about 5% for both exactly and partially matched alternatively–lexicalized discourse signals due to weak supervision.

1 Introduction

Understanding the underlying structure of a text is a fundamental problem in computational linguistics. In discourse analysis, shallow discourse parsing in particular, we aim to identify individual discourse relations within a text. Thus we can gain information that helps in downstream tasks such as automatic summarization, machine translation, and document classification. The study of connecting phrases not only helps in understanding the way people connect their thoughts but also in the identification of discourse relations anchored by them. For our work, we use the third version of the Penn Discourse Treebank (PDTB) (Prasad et al., 2018) that distinguishes between explicit relations (signaled by a closed set of discourse connectives, e.g. because, and, if-then, and before) and alternative lexicalizations (signaled by connecting phrases other than discourse connectives, e.g. this means, for that reason, and it all adds up to). In total, the PDTB contains 25878 signaled relations, most of which belong to the group of explicit relations (94%). Only 1638 connecting phrases build the group of free connective phrases, in the corpus referred to as alternative lexicalizations. While explicit relations are more commonly used to verbalize expansions and comparisons between text spans, alternative lexicalizations often point to lexically grounded causal relations. Also, they potentially contain information, e.g. the phrase the most crucial reason for that gives also evidence about the reason’s importance, which is useful for understanding the full discourse.

In our work, we aim to overcome the problem of very limited training data available for free connective phrases and examine a weakly-supervised scenario for continuously improving a model through its own predictions. We regularize these predictions by re-ranking the extracted signals through a separate model trained to discriminate possible signal candidates into signals with or without discourse usage. Summarized, our contributions are: We (i) present a novel architecture and provide state-of-the-art results for recognizing alternative lexicalizations in the recent version of the PDTB. Further, we (ii) improve its performance of recognizing phrases by integrating unlabeled data into the training process using weak supervision.

2 Related Work

Self-supervised learning (Yarowsky, 1995), the most simple semi-supervised learning algorithm, extends its training data by adding new samples with confident predictions on different data. Self-training has been successfully applied on constituent parsing (McClosky et al., 2006) by incorporating a re-ranking strategy (Charniak and Johnson, 2005; Collins and Koo, 2005) to improve parsing results and reduce the bias of the trained model. Also, Suzuki and Isozaki (2008) improved performance on part-of-speech tagging via sequence labeling. In recent work, Nishida and Matsumoto (2022) study the empirical effectiveness of bootstrapping annotations from out-of-domain data and...
show its positive impact for BERT-based discourse dependency parsers. For candidates selection, they study criteria inspired by Steedman et al. (2003).

Chou et al. (2014) approach semi-supervised learning for named entity recognition, a similar training problem (sequence labeling) as ours. They propose an additional model for estimating confidence (self-testing) and removing samples with low scores. Braud et al. (2016) first apply semi-supervised learning to RST discourse parsing using multiple views on the data by incorporating various auxiliary tasks, such as PDTB discourse parsing. Knaebel and Stede (2020b) improved their argument extraction by jointly training three separate models so-called tri-training on additional news documents. Recently, Kobayashi et al. (2021) successfully bootstrapped RST sub-trees using a combination of simpler feature-based teachers to train a more complex neural student.

The group of alternative lexicalized relations has been rarely studied. Prasad et al. (2010) did initial work on the identification and analysis of alternative lexicalized relations in an older PDTB version. Synková et al. (2017); Rysová and Rysová (2015) distinguished two classes of alternative lexicalizations and developed a dictionary approach for more regular alternative lexicalized phrases. Most recently, Knaebel and Stede (2022) implemented the first automatic neural-based model for recognizing alternative lexicalizations on a sentence level using a binary sequence labeling approach. In this work, we build on their approach and adapt this model to the paragraph level, similar to the explicit connective model of Kurfalí (2020).

3 Method

3.1 Recognizing alternative lexicalizations

The recognition of alternative-lexicalized discourse signals (AltLex) in the PDTB corpus is challenging due to the higher complexity of the phrases when compared to explicit signals for example, and the limited number of training samples. While Knaebel and Stede (2022) predict binary labels (is-part-of the signal) on the sentence level, we follow Kurfalí (2020) and integrate more context into the model by training the whole model on the paragraph level. Accessing more context seems unavoidable for improving performance as discourse signals naturally link to phrases outside their sentence. We make use of pre-trained large language models and fine-tune the base model combined with an additional token classification layer on top of it.

Shifting from sentences to paragraphs results in potentially having an arbitrary number of signals. For this purpose, we use a three-class encoding similar to Kurfalí (2020): single signals, e.g. following, resulting, not, and soon, multi-word signals, e.g. for this reason and in addition to, and no signal otherwise. We limited our experiments to continuous signals, e.g. we removed phrases like the more [. . .], the more, which removes a minor number of samples but allows for decoding the labeled sequence without redundancy. We did not choose a more complex signal encoding, such as BIOS and BIOES, due to the lack of available training data and the resulting class imbalance.¹

3.2 Learning from unlabeled data

In this work, we study self-training, which is a very basic but effective semi-supervised learning technique that uses a model’s self-estimation to integrate confident predictions from unlabeled data. However, this technique has a high bias due to reinforcing its own false predictions. We overcome this problem by, first, improving the base performance of our signal extractor by building an ensemble of three separately trained models. Second, we follow the idea of McClosky et al. (2006) and introduce a separate model for confidence estimation that not only reduces the bias of a singly self-trained model but also simplifies the determination of a confidence score.

To estimate the model’s confidence in its predicted alternatively-lexicalized phrases, referred to as candidates, we design an auxiliary task to disambiguate signal candidates produced by the labeling model. We want to learn to discriminate candidate phrases into those related to an AltLex or not. We adapt previous work on explicit sense classification (Knaebel and Stede, 2020a) to alternative lexicalizations and simultaneously predict whether a possible candidate phrase is used as a discourse signal and if so, we learn to predict its sense. Instead of learning only a single sense level, we jointly learn sense versus no-sense prediction on coarse and fine senses as Long and Webber (2022) suggest in their work. In a short ablation study (see Appendix A), we show that our chosen disambiguation architecture works with similar performance as a simple binary classifier.

¹We did some initial studies with BIOS and BIOES encodings, but the performance was not satisfying.
Our self-learning approach consists of two phases (compare Figure 1). In Phase 1 (Recall), we optimize our signal labeling model (Signal Extraction) with respect to high recall. Therefore, we lower the weight for the None class, which is the dominating class label, and thus implicitly reinforce a higher focus on the other class labels. The resulting increase in the recall value simultaneously leads to a reduction in the precision value of the model. We first extract signal candidates from the PDTB which results in a fuzzy version, in order to train a second model for signal disambiguation (Signal Classifier). In the final step of a single iteration, we extract signals from a different corpus (here NYT see Section 4.1) and filter these signals, based on the confidence score, before we use the confident paragraph samples together with the original training data. Confident paragraphs are defined based on the individual signal confidence, such that signals are removed if the confidence is below a relation threshold $\tau_{rel}$ and the remaining signals’ minimum is higher than a paragraph threshold $\tau_{par}$. In Phase 2 (F1Score), we use the finally extracted confident paragraphs and train a new model on signal extraction, but this time all class labels are equally weighted and the model is optimized for F1 score.

4 Experiments

4.1 Experimental Settings

The Unlabeled Corpus Most, but not all (Webber, 2009), documents of the PDTB are news articles. To learn about alternative lexicalizations from a different corpus, as there is relatively little annotated data currently available, we choose another news wire corpus, under the assumption of an easier adaption of a similar domain compared to other genres. The New York Times Annotated Corpus 2, referred to as NYT, contains about 1.8 million documents published by the New York Times between 1987 and 2007. For our work, we use a random subset of documents, 200 per month from the years 2000–2002, sampled only once before the experiments. The reduced corpus is due to computational feasibility, the years were selected randomly. We selected NYT to complement the PDTB training data because much more data is available and it has similar quality and structure of articles as in the Wallstreet Journal corpus, which is used for the PDTB. For example, we decided against the CNN/DM corpus used in a different study (Kobayashi et al., 2021) because of the largely absent paragraph structure.

Hyper-Parameter Settings For data preparation, we split 10% of documents from the PDTB corpus for testing purposes. While we use varying test splits for the general evaluation of the architecture, also to compare to previous work, we use the same test split for the evaluation of the self-supervised setting. After creating a separate test set, in each run, we split another 10% of the remaining training documents for validation. To increase the reproducibility of our experiments, we use the same validation splits for each model run, e.g. we have the same 3 and 5 splits for model ensembles and evaluations, respectively. For both types of models, signal labeling and sense classification, the batch size is 32. We train for at most 10 epochs and stop after 3 epochs without any improvement. For optimization, we chose Adam with decoupled weight decay (Loshchilov and Hutter, 2019) and an initial learning rate of $1e^{-4}$ that is reduced linearly over the maximum training epochs. As we observed overfitting with a too-small dropout rate, we set it to 0.4 for both models. For embedding paragraphs, we chose the base architecture of RoBERTa (Liu et al., 2019) that has shown good performance on several other tasks related to discourse processing (Long and Webber, 2022; Koto et al., 2021; Guz et al., 2020). We fix all but the last two layers.
for signal labeling. For the disambiguation model, we extract all hidden units from the model and propagate them to our classifier. As the input size of RoBERTa is limited, we truncate the tokenized paragraph. Only less than 1% of the paragraphs are affected by this truncation. For signal classification, we remove training examples where a signal occurs after the limit.

During the adaption phase, we focus on the recognition of alternative lexicalizations rather than whether predictions are correct or not, as we later train an additional model that filters wrong predictions. We identify two crucial hyper-parameters: First, we examine changing the **majority class weight** (None class) for the cross-entropy loss. Second, we study the influence of **negative samples** on the training progress. In agreement with our experiments (for details see Appendix B), we chose 0.01 for the majority class weight as the next step’s small increase in recall did not justify the higher decrease in precision. Further, our results indicate that there is no advantage in reducing the number of negative samples.

For both phases, we set the relation threshold \( \tau_{rel} \) to 0.33 as we measured a good balance of true and false predictions on the PDTB data. For the paragraph threshold \( \tau_{par} \) we use a value of 0.7 during training, as we focus on optimizing the recognition rate (recall) of the extraction model in this phase. In the second phase, we study varying thresholds ranging from 0.4 to 0.9 for minimal paragraph relation confidence.

### 4.2 Experimental Results

First, we evaluate our novel architecture and compare its base performance with the initial work by Knaebel and Stede (2022). In their work, they measure the overlap within sentences containing an alternative lexicalization. We, therefore, re-run their neural labeling model and use the same evaluation metrics (exact–match) as for this paper. Results are averaged over 10 random splits and presented as mean (M) and standard deviation (SD). In our evaluation under similar conditions, the baseline (M=34.07% F1, SD=6.09) is clearly outperformed by our introduced model (M=45.48% F1, SD=5.08). We also study the performance of ensembles as used in our self-learning setting and simply combine the output probabilities of three random models. The performance further improves (M=51.68% F1, SD=3.28) and we observe a decreasing standard deviation.

Results of our final experiments are shown in Figure 2 and in more detail in Appendix D. We compare the baseline trained on the original PDTB dataset with models of varying paragraph thresholds \( \tau_{par} \) (0.4 to 0.9) that incorporate data from the NYT corpus into their training data. We utilize partial matching as introduced by Xue et al. (2016), and define the matching overlap based on the F1 score of two connecting phrases. **Partial-Match** and **Exact-Match** refer to 70% and 90% F1 matching thresholds, respectively. For example, our model recognizes two of three words of the signal *greatly expanding collaboration* correctly, resulting in 0.66 recall, 1.0 precision, and thus 0.83 F1, this signal would count as partially matched but not exactly. All Experiments run on the same test set, with varying training and validation splits, 5 repetitions each. Interestingly, all models perform best at a \( \tau_{par} \) of 0.6, which is in accordance with the threshold suggested by Nishida and Matsumoto (2022). Our model (M=47.38% F1, SD=1.22) with all unlabeled data and \( \tau = 0.6 \) improves the baseline (M=42.95% F1, SD=2.52) by more than 4% F1 score on exact match.

### 4.3 Analysis of Selected Cases

In this section, we would like to show some selected signal examples that we noticed while reviewing the results. First, we look at the predictions of our recall-optimized signal extraction model (without filtering predictions by our second classifier) within the PDTB training data. This model has repeatedly recognized phrases (after, and, on the other hand, at the same time, further, if, because [of], among others) as alternative lexicalizations although, in terms of their surface form, they should rather belong to the group of explicit connectors. We assume some of these phrases are only partially recognized alternative phrases e.g. signals in which the referential expression is missing after this situation and because of that event. We also identify cases where individual parts of the signal belong to explicit connectives, while their conjunction is rather considered as an alternative lexicalization, e.g. *since and then*. Despite a large number of possible explicit signals, most of the confused signals are filtered in the second step and are therefore not considered signals at all. Interestingly, we noticed that the model identifies a few signals at the beginning of a paragraph, similar as discussed by Prasad.
et al. (2017), that are per definition not included in the PDTB annotations, e.g. That explains why, To illustrate, All this illustrates that, and What’s more. Besides different variations of gerunds, we found phrases such as at the most, that may mean, and even that that are likely being used as discourse signals without checking their context. The integration of the model’s output in future annotation processes may be beneficial in identifying more discourse signals.

Next, we examine the predicted alternative lexicalizations in the NYT data. Here, we found quite a few change verbs, e.g. diminishing, bolstering, stimulating, absorbing, contributing and negating, that never occurred in the training data. There is about the same number (20 each) of variants of alternative lexicalizations containing the word reason in both data sets which have no overlap with the respective other data set, e.g. the reason is probably that, the reasons for that finding, and that alone is reason for where identified in NYT but not in PDTB. We further found, that our model tends to predict shorter signals (average length of 9) compared with the PDTB training dataset (average length of 13). The longest extracted signals with respect to token counts are one reason for the cooperative ads is that, the overhaul was spurred in part by, and and that might partly explain why.

5 Discussion and Conclusions

We developed a new paragraph-based architecture to extract alternatively-lexicalized discourse signals and presented state-of-the-art performance. Initial experiments on incorporating non-annotated data showed a further increase in performance. Size seems to matter for this learning too, as this principle often holds for deep learning models. Although the gaps are rather small for up to 10,000 sampled documents, we think the distance for the largest set of documents is very clear. Due to time and computation constraints, we could not identify an upper performance bound yet.

We notice throughout our signal extraction experiments a confusion between alternative lexicalizations and explicit connectives. We assume the model to have problems clearly understanding their difference, as both kinds of phrases signal discourse relations. Filtering the connecting phrases as we have done seems unavoidable. Contrary to this, however, it seems worthwhile to soften the boundaries between these two categories and develop models that combine both types. This is not trivial due to the differences between both signal types (explicit signals are usually shorter recurring phrases with higher frequencies; AltLex signals tend to be longer phrases with more variance).
Limitations

Although the new architecture works well on PDTB-like structured data, we are often challenged with texts without clear paragraph structure. This would make it either necessary to pre-process texts and split sentences into semantically closed paragraphs such that our proposed model takes advantage of the surrounding context, or develop a new sentence-based model which was not successful in previous work.

Limiting the model to predict only continuous alternative lexicalizations does not highly affect results on the PDTB, but might have a more considerable impact on other text genres, e.g. speeches and debates. This would require the use of a more complex signal encoding as mentioned in Section 3.1.

Acknowledgements

This research has been supported by the German Research Foundation (DFG) with grant number 455911521, project "LARGA" in SPP "RATIO". I would like to thank Manfred Stede and Freya Hewett as well as the anonymous reviewers for their valuable feedback.

References


Table 1: Ablation Study for the AltLex Candidate Classifier. Results show mean and standard deviation for 10 runs each.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>83.54 (5.99)</td>
<td>59.45 (12.34)</td>
<td>68.84 (9.68)</td>
</tr>
<tr>
<td>Coarse</td>
<td>86.32 (4.83)</td>
<td>51.78 (9.22)</td>
<td>64.36 (7.82)</td>
</tr>
<tr>
<td>Binary</td>
<td>85.18 (4.81)</td>
<td>56.81 (8.80)</td>
<td>67.80 (6.87)</td>
</tr>
</tbody>
</table>

A Candidate Disambiguation: Ablation Study

Discourse signal disambiguation is a fundamental step in our weakly-supervised learning cycle for improving the prediction quality of our signal extraction model. We intuitively followed previous work on signal-based sense classification (Knaebel and Stede, 2020a) with the assumption of better results learning multiple sense levels at once. (Long and Webber, 2022) Our ablation study in Table 1 shows, that contrary to our assumption the baseline and a binary classifier that is limited to predicting the discourse usage of a free connective phrase have similar performances. Removing the model’s fine-sense classification drastically reduces the recall of identified signals but increases precision. This holds for the binary case, too. Further investigations are necessary to identify specific differences in these classifiers.

B Hyperparameters: Loss Weight and Negative Sampling

During the adaption phase, we focus on the recognition of alternative lexicalizations rather than whether predictions are correct or not, as we later train an additional model that filters wrong predictions. Therefore, we adjust the majority class weights (None class) of the cross-entropy loss. In Table 2a, we report macro averaged results for weights ranging from 1.0 (normal weight) to 0.001 (inverse occurrence weight). As expected, the results indicate an increase in average recall with a decrease in average precision at the same time. We chose 0.01 for the majority class weight as the next step’s small increase in recall did not justify the higher decrease in precision.

We also study the influence of negative samples on the training progress. The results in Table 2b indicate no advantage of reducing negative samples for training data, as already mentioned so in the paper. However, in contrast, a broader study with varying test partitions showed an increase in recall while reducing the number of negative samples.

C Numbers of Extracted Paragraphs

Table 3 summarizes the number of training samples that were extracted from a given corpus sample (limited by the number of documents) and a corresponding relation paragraph threshold that needs to be satisfied for positive training samples.

D Full Final Results

Table 4 summarizes our final experiments’ results in full detail. Partial-Match and Exact-Match refer to 70% and 90% overlap, respectively. In contrast to the evaluation with previous work, for this evaluation, we split test data only once at the very beginning and stay with it throughout the evaluation. Results are averaged over different validation splits, though.
Table 4: Full results, partial and exact matching, of final model with varying paragraph threshold (0.4 to 0.9) trained on data including NYT corpus. All experiments run on the same test set, with varying training and validation splits, results are averaged over 5 repetitions.
Entity-based SpanCopy for Abstractive Summarization to Improve the Factual Consistency

Wen Xiao and Giuseppe Carenini
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
{xiaowen3, carenini}@cs.ubc.ca

Abstract

Discourse-aware techniques, including entity-aware approaches, play a crucial role in summarization. In this paper, we propose an entity-based SpanCopy mechanism to tackle the entity-level factual inconsistency problem in abstractive summarization, i.e., reducing the mismatched entities between the generated summaries and the source documents. Complemented by a Global Relevance component to identify summary-worthy entities, our approach demonstrates improved factual consistency while preserving saliency on four summarization datasets, contributing to the effective application of discourse-aware methods summarization tasks. \(^1\)

1 Introduction

Discourse-aware models play a crucial role in natural language processing applications, including machine translation (Guzmán et al., 2014) and text summarization (Xu et al., 2020). Among these applications, abstractive text summarization, the task of generating informative and fluent summaries of the given document(s), has attracted much attention in the NLP community. While early neural approaches focused more on designing customized architectures or training schema (Nallapati et al., 2016; Tan et al., 2017; Liu* et al., 2018), recent works have shown that both pre-trained generation models fine-tuned on in-domain datasets and zero-shot GPT-like decoder-only models generally have better performance (Lewis et al., 2020; Zhang et al., 2020; Raffel et al., 2020; Goyal et al., 2022).

However, even with state-of-the-art performance on standard automatic evaluation metrics such as ROUGE (Lin, 2004) and BERTScore (Zhang* et al., 2020), the generated summaries still suffer from the problem of factual inconsistency, which means the generated summaries may not be factually consistent with the content expressed in the source documents (Kryscinski et al., 2020; Bubeck et al., 2023). Inconsistencies may exist either at the entity, where summaries mention entities absent from source documents, or at the relation level, where summaries express relations between entities that differ from the source (Nan et al., 2021).

In this paper, we focus on the entity-level inconsistency problem, i.e., to make the model generate summaries with less entities which do not appear in the source document(s) i.e., ‘hallucinated’ entities. Note however, that hallucinated entities are not necessarily ‘unfaithful’ or ‘wrong’ (Cao et al., 2021), so the goal is to reduce them without excluding entities that do appear in the reference summary i.e., without penalizing saliency. Table 1 shows an example of entity-level factual inconsistency from the XSum dataset. The summary generated by PEGASUS totally missed one entity (Royal Marine) and one entity indicates a larger area than the correct one (Hampshire).

<table>
<thead>
<tr>
<th>Entities in Source Doc:</th>
<th>Royal Marine, Falklands, Portsmouth, Falklands War Memorial,...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth:</td>
<td>Plans to move a statue depicting a Royal Marine in the Falklands conflict away from Portsmouth seafront have been criticised.</td>
</tr>
<tr>
<td>PEGASUS:</td>
<td>A campaign has been launched to keep a statue of a Falklands War marine in Hampshire.</td>
</tr>
<tr>
<td>SpanCopy:</td>
<td>A campaign to keep a statue of a Royal Marine marching across the Falklands in Portsmouth has been launched.</td>
</tr>
<tr>
<td>SpanCopy + GR:</td>
<td>A statue of a Royal Marine marching across the Falklands during the Falklands War Memorial should remain in its current location, campaigners have said.</td>
</tr>
</tbody>
</table>

Table 1: An example of entity-level factual inconsistency from the XSum dataset. The summary generated by PEGASUS totally missed one entity (Royal Marine) and one entity indicates a larger area than the correct one (Hampshire).
example of entity-level factual inconsistency from the XSum dataset. Although the content of the summary generated by the SOTA summarizer PEGASUS (Zhang et al., 2020) is roughly similar that of the ground-truth summary, it does not accurately summarize the original documents with the proper entities. Specifically, the entity ‘Hampshire’ is ‘hallucinated’, as it does not appear in the source document. Despite the fact that the city ‘Portsmouth’ is located in ‘Hampshire’ county, the entity itself is still an instance of factual inconsistency (i.e., an unnecessary generalization).

Prior work (Dong et al., 2020; King et al., 2022) mainly address the entity-level inconsistency problem in the post-processing stage. However, those methods either requires additional sophisticated models, e.g. Dong et al. (2020) uses a pre-trained QA model to ‘revise’ the generated summaries, or being built on arguably brittle heuristics (King et al., 2022). Recent work (Nan et al., 2021) proposes two ways to directly improve the end-to-end summarization model, either by training with an auxiliary task, which is to recognize the summary-worthy entities in the source document using the hidden states from the encoder, or jointly generating the entities and the summaries, i.e. generating a chain of entities in the summary followed by the summary. The latter one is in-line with recently proposed entity-aware guided summarization methods (He et al., 2020; Narayan et al., 2021). Yet, both methods do not explicitly encourage the model to generate the summaries with more valuable entities, as both of them aim to guide the model to detect the summary-worthy entities without any changes in the summary generation process. Instead, aiming for a lean and modular solution, we propose the discourse-aware SpanCopy Mechanism to explicitly copy the matched entities\footsup{2} from the source documents when generating the summaries. One key advantage of our proposal is that it can be easily integrated into any pre-trained generative sequence-to-sequence model.

Since often only a few of the entities in the source documents can be included in the summary, which we call ‘summary-worthy entities’, we also explore an additional Global Relevance component to better recognize the summary-worthy entities by automatically generating a prior distribution over all the entities in the source documents.

\footsup{2}We particularly focus on the Named Entities in this paper, but our method can be easily applied to any kind of spans or entities.

We test our proposal on four summarization datasets in the news and scientific paper domain, comparing it with the established SOTA PEGASUS system (Zhang et al., 2020). In a first set of experiments, as a sanity check, we assess our models on arguably easier subsets of these datasets, where all the entities in the reference summaries belong to the source document. In these cases, SpanCopy should definitely dominate PEGASUS, which is confirmed by the results. In a second set of experiments, we fine-tune and test on the full datasets. On this realistic and more challenging task, we find that SpanCopy (without Global Relevance) can strongly improve the entity-level factual consistency (+2.28) on average across datasets, with essentially no change in saliency (−0.06).

2 Related Work

2.1 Abstractive Summarization

Early neural abstractive summarization models (Nallapati et al., 2016; Paulus et al., 2018; See et al., 2017) are mainly sequence-to-sequence models based on different variants of RNN, e.g. LSTM or GRU, with additional components targeting different properties of the summaries, like redundancy (Tan et al., 2017) and coverage (See et al., 2017). However, all the recurrent models suffer from serious weakness like long-term memory loss, and requiring excessive time to train.

To tackle these problems, researchers in the area of abstractive summarization started to use attention-based transformer models (Liu and Lapata, 2019a,b); recently reaching SOTA performance when pre-trained generative transformers are applied to the task, e.g. BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020) and PRIMERA (Xiao et al., 2022). The SpanCopy mechanism we propose in this paper can be advantageously injected into any pre-trained models.

2.2 Factual Consistency

Despite the large improvements with respect to automatic evaluation metrics, recent studies (Cao et al., 2018; Kryscinski et al., 2020) show that around 30% of the summaries generated by the SOTA summarization models contain factual inconsistencies. Ideally, the assessment of factual consistency should rely on human annotations (Maynez et al., 2020), but these are costly, time consuming and lack a unified standard. Thus promising automatic evaluation metrics for factual consisten-
cies of generated summaries have been explored in recent years. To assess relation-level factual consistency two kinds of metrics have been proposed: one based on classification (Kryscinski et al., 2020), and one based on Question-Answering (Maynez et al., 2020; Durmus et al., 2020). For entity-level factual consistency, the focus of this paper, Nan et al. (2021) propose a simple but effective evaluation metric, based on the matched named entities in both generated and ground-truth summaries. In our work, we use such metric to evaluate whether the generated summaries are consistent with both the source documents and the reference summaries at the entity-level.

2.3 Entity-aware Summarization

The use of entities as part of discourse-aware approaches has been shown to improve both saliency and factual consistency for the summarization task. Xiao et al. (2022) identify salient entities within document clusters and utilize them to select pseudo summaries during the pre-training phase, leading to superior performance on multiple datasets. Entities have also been employed in guided summarization, where researchers extract oracle entities from ground-truth summaries and use them to guide summary generation. For instance, Dou et al. (2021) introduce an additional encoder to encode guidance signals, sharing partial parameters with the original document encoder. In related research, He et al. (2020) propose a pre-training strategy that prepends source documents with oracle keywords as prefixes, and Narayan et al. (2021) train models to first predict an entity chain before generating the final summary. Diverging from prefix-based strategies, our approach enables the model to learn explicit copying of entities to specific positions in the generated summary, further advancing discourse-aware summarization techniques.

2.4 Copy Mechanism

See et al. (2017) first apply pointer-generator network in an abstractive summarization model, which facilitates copying words from the source documents by pointing, i.e., generating a distribution of probabilities to copy each word from the source. Following their work, Bi et al. (2020) propose PALM, in which the copy mechanism is applied on top of the transformer model, and with a novel pre-training schema, the model achieves SOTA on several generative tasks, such as abstractive summarization and generative QA. More recently, Li et al. (2021) further explores how to make use of the copy history to predict the copy distribution for the current step. However, all the aforementioned works focus on copying at the word level, which tends to be sparse and noisy. Instead, we aim to train the model to copy spans of text i.e., the named entities, in this paper.

Admittedly, some previous work has also investigated span-based copy mechanisms. Yet, those models either predict the start and end indices of a span (Zhou et al., 2018), or predict the BIO labels for each token (Liu et al., 2021). Even if such...
strategies can copy any kinds of spans (clauses, n-grams, entities, phrases or longest common sequence) from the source document, they may introduce unnecessary noise and break the coherence of the generated text. In this work, we focus on copying the spans of the Named Entities, extracted by a high-quality NER tool, aiming to improve factual consistency of the generated summary without negatively affecting saliency.

3 Our SpanCopy Method

3.1 Transformer-based Summarizers

Typically, transformer-based summarization (Lewis et al., 2020; Zhang et al., 2020) consists of two steps (i) The Encoding Step (by the Encoder shown in yellow in Fig.1), which encodes the source input(s) into an hidden space; (ii) the Decoding Step, which computes a probability distributions on the output vocabulary to generate each token of the resulting summary. In this paper, to better describe our methods in the context of a generic summarization models, we split the Decoding process into two components, the Decoder itself (shown in green in Fig.1), which outputs the representations of predicted tokens, and the Generator (shown in purple in Fig.1), an MLP layer mapping the representations to the final probability distribution on the output vocabulary.

More formally, for a document with n tokens $D = \{t_1^d, t_2^d, ..., t_n^d\}$, and the corresponding summary with m tokens, $S = \{t_1^s, t_2^s, ..., t_m^s\}$, the output of the Encoder is a sequence of hidden states of all the tokens, i.e. $\{h_1^e, h_2^e, ..., h_n^e\}$. And then the Decoder predicts a sequence of vector, $\{h_1^d, h_2^d, ..., h_m^d\}$, representing the tokens to be predicted. Finally, the Generator maps those vectors to the distributions over the vocabulary, i.e. $\{p_1, p_2, ..., p_m\}$, where $p_i \in \mathbb{R}^{|V|}$.

There has been recent research on models with a decoder-only structure for summarization (Goyal et al., 2022), where the decoder is responsible for both the encoding and decoding steps. In this approach, the tokens in the source documents are represented using the output of the decoder, rather than relying on the encoder. Our proposed method specifically applies to the decoder, making it compatible with both encoder-decoder models and decoder-only models. However, in this paper, we primarily focus on exploring the application of our method to encoder-decoder models, while leaving the investigation of decoder-only models for future research.

3.2 SpanCopy Mechanism

A key problem with generic sequence-to-sequence transformer-based summarizers is that the decoding step is prone to generate factual inconsistencies, i.e. the model may make up entities or relations that are not entailed by the source documents. To address entity-level factual inconsistency, we introduce in the Decoding Step the SpanCopy mechanism, which can be conveniently plugged into any pre-trained models. Specifically, we first identify and match the entities in both source document and summary, and then instead of generating the entire summary word by word, we add an additional Span Copier to directly copy entities from the source document, with a Copy Gate predicting the likelihood of whether the model should generate the current token from the vocabulary or directly copy an entity from the source document.

**Span Copier** (shown in blue in Fig.1) is an attention module over all the entities in the input document. Suppose there are $|E|$ entities in the input document, with each entity $j$ being a span over tokens $[d_{jx}, d_{ju}]$, then the entities can be simply represented as $e_j = \text{avg}([h_{jx}^e : h_{ju}^e])$, where $h_e^j$ represents the output of the encoder for each token $d_i$. At each decoding step $i$, we compute the logit vector of copying each entity at the current step as:

$$o_i^e = Q(h_i^d) \cdot K(e_j), o_i^e \in \mathbb{R}^{|E|}$$ (1)

indicating how likely it is to copy the entities from the source document at each step. Notice that to better balance the numeric difference caused by the size of selection space ($|V|$ and $|E|$), we generate and combine the raw logit vectors\(^3\) from the Span Copier and Generator, and take softmax over the combined space to get the final probability.

**Copy Gate** (shown in red in Fig.1) is a classifier to map the hidden states to a singular value, i.e.

$$p_i^{\text{copy}} = \sigma(MLP(h_i^d)), p_i^{\text{copy}} \in [0, 1]$$ (2)

which indicates the probability of copying an entity at each step. On the contrary, $1 - p_i^{\text{copy}}$ represent the probability of generating a token from the vocabulary at step $i$.

\(^3\)The vector of raw (non-normalized) predictions that the classification model generates...
Then the final probability, combining both generation over the vocabulary and the copy mechanism over the entity space, is computed as

\[
p^{\text{final}}_i = \text{softmax}(\{1 - p^{\text{copy}}_i\} \cdot o^g_i + p^{\text{copy}}_i \cdot o^c_i)
\]  

(3)

with \(p^{\text{final}}_i \in \mathbb{R}^{(|V|+|E|)}\), where \(o^g_i \in \mathbb{R}^{(|V|)}\) is the logit vector of token generation and \(o^c_i \in \mathbb{R}^{(|E|)}\) is the logit vector of entity copying. As a result, the first \(|V|\) dimensions of the final probability represent the probability of generating all the tokens from the vocabulary, while the following \(|E|\) dimensions contain the probabilities of copying the entities from the source document.

Note that the input of the original Decoder in the transformer model at each step is the embedding of the previous token (which is the ground-truth one during training, and the predicted one for inference), but a span of text longer than 1 does not naturally have an embedding to match. We simply use the average of the embedding of all the tokens in the entity, following previous work using average embedding to represent a span of text (Xiao and Carenini, 2019).

3.3 Loss

We use the standard loss for abstractive summarization, i.e. the cross entropy loss between the predicted probability and the ground truth labels,

\[
L_1 = \sum_i L_s(p^{\text{final}}_i, t_i)
\]  

(4)

However, notice that, since the predicted probability distribution is over the combined space of vocabulary size and entity size \(p^{\text{final}}_i \in \mathbb{R}^{(|V|+|E|)}\), the corresponding ground truth labels can be either indices of words to be generated from the vocabulary, or the indices of entities to be copied from the source document, i.e. \(t_i \in [0, |V| + |E|]\). Specifically, if \(t_i < |V|\), then the \(t_i\)-th token should be generated, and if \(t_i > |V|\), the \((t_i - |V|)\)-th entity should be copied from the source document.

3.4 SpanCopy with Global Relevance

Among all the entities in the source documents, there are only a few summary-worthy entities that should be copied into the summary (e.g. around 10% in CNN/DM and 1.5% in arXiv). To make the model better recognize such summary-worthy entities, we explore a Global Relevance (GR) component, which takes all the entities in the source document as inputs, and predicts how likely each entity is to appear in the final summary. We use the generated ‘entity likelihood’ as a prior distribution for the Span Copier component, with GR also trained as an auxiliary task.

Global Relevance is a classifier mapping the hidden state of a source document entity into a value within \([0, 1]\), indicating the probability that such entity should be included in the summary.

\[
gr = \sigma(MLP(e)), gr \in \mathbb{R}^{|E|}
\]  

(5)

Then \(p^{\text{final}}_i\) in Eq.3 is updated with \(gr\) as

\[
p^{\text{final}}_i = \text{softmax}(\{1 - p^{\text{copy}}_i\} \cdot o^g_i + p^{\text{copy}}_i \cdot o^c_i \cdot gr_i)
\]  

(6)

New Loss As an auxiliary task, we also train the model with the ground-truth GR labels to make it more accurate. Specifically, the label \(y^{gr}_i = 1\) if the \(i\)-th entity in the input document is included in the ground truth summary. Then we update the loss function with \(L_{gr}\) balanced by \(\beta\):

\[
L_2 = (1 - \beta) \sum_i L_s(p^{\text{final}}_i, t_i) + \beta \sum_j L_{gr}(gr_j, y^{gr}_j)
\]  

(7)

4 Experiments and Analysis

4.1 Settings

SpanCopy can be plugged into any pre-trained generation model. In this paper, we use PEGASUS(Zhang et al., 2020) as our base model, since it has delivered top performance on multiple summarization datasets. We recognize named entities with an off-the-shelf NER tool4. The balance factor \(\beta\) of GR is set by grid search on small subsets of each dataset (2k for training and 200 for validation).

4.2 Evaluation Metrics

To evaluate the saliency and entity-level factual consistency of the generated summaries, we apply the following metrics:

Saliency metrics assess the similarity of the generated summary with the reference summary. 

ROUGE scores (Lin, 2004) measure the n-gram overlaps between generated and ground truth summaries. We apply the metrics R-1, R-2 and R-L.

Summary-precision, -recall and -f1 (sum_p, sum_r and sum_f) (Nan et al., 2021) measure the

4https://spacy.io/
We test and compare our SpanCopy model with XSum(Narayan et al., 2018) and scientific papers (Table 2: Statistics of all the datasets (original/filtered), on the lengths \((L_{doc}, L_{summ})\) and number of entities \((N_{doc}, N_{summ})\) in the source documents and ground truth summaries, as well as \(src_p(gt)\), the entity level source-precision of the ground-truth summary.

### Entity-level factual consistency metric:
Measures the named entity matching between the generated summaries and the source documents. (Nan et al., 2021) With \(NE(D)\) and \(NE(S_{gen})\) representing the named entities in the source document and generated summaries, respectively, \(src_p = \frac{|NE(S_{ref}) \cap NE(S_{gen})|}{|NE(S_{gen})|}\) measures how many entities in the generated summaries are from the source documents, i.e. \(src_p = \frac{|NE(D) \cap NE(S_{gen})|}{|NE(D)|}\). It is an evaluation metric for entity-level factual consistency, as it directly measures how consistent the generated summaries are with the source.

### 4.3 Datasets
We test and compare our SpanCopy model with the original PEGASUS on four datasets, in the domains of news (CNNDM(Nallapati et al., 2016), XSum(Narayan et al., 2018)) and scientific papers (Pubmed and arXiv(Cohan et al., 2018)). As a sanity check, we initially assess our models on subsets of these datasets, where all the entities in the reference summaries belong to the source document (we call these filtered datasets). In these cases \((src_p(gt) = 1)\), Span Copy and GR should dominate PEGASUS, because by design they tend to generate entities from the source document. \(^5\)

The statistics of the filtered and original datasets, on the lengths and number of entities in the document and summaries, can be found in Table 2. \(src_p(gt)\) measures the entity-level factual consistency between the source document and the ground-truth summary, with lower value meaning that there are more novel entities in the ground-truth summaries. The table shows that the datasets in the news domain have higher density of the entities with respect to the lengths (number of words) of both documents and ground-truth summaries, i.e. \(N_{doc}/L_{doc}\) and \(N_{summ}/L_{summ}\) are larger for the news articles. A possible explanation is that news articles tend to describe an event or a story, which may contain more names of people, organizations, locations, etc., as well as dates. Interestingly, CNNDM and Pubmed contain less novel than the other two datasets (with higher \(src_p(gt)\)), something that the proposed SpanCopy mechanism may benefit from. Comparing the filtered datasets with the original ones, we can see that the number of entities in the summaries drops for all the datasets, especially for arXiv, as the more entities in the summary, the less likely they can be all matched to the source documents.

### 4.4 Results and Analysis
The results on the filtered and original datasets are shown in Table 3 and Table 4.

#### Filtered Datasets
We first evaluate our models, with the backbone model, PEGASUS on the filtered datasets, which is an easier task, and the results can be found in Table 3. All the models are fine-tuned and tested on the filtered datasets. Since we only keep the examples with all the entities in the summaries being matched with the entities in the source documents, the theoretical ceiling of

\(^5\)Statistics of the (filtered/original) datasets can be found in Appendix.B
<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE</th>
<th>Entity(Summ)</th>
<th>Entity(Doc)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-2</td>
<td>R-L</td>
</tr>
<tr>
<td>CNNDM Filtered</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>44.70</td>
<td>22.23</td>
<td>32.52</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>45.46</td>
<td>23.12</td>
<td>33.48</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>45.74</td>
<td>23.44</td>
<td>33.67</td>
</tr>
<tr>
<td>XSum Filtered</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>43.01</td>
<td>19.00</td>
<td>35.21</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>43.23</td>
<td>19.90</td>
<td>35.50</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>43.78</td>
<td>19.12</td>
<td>34.97</td>
</tr>
<tr>
<td>Pubmed Filtered</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>46.99</td>
<td>21.46</td>
<td>42.57</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>44.23</td>
<td>22.34</td>
<td>43.43</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>43.78</td>
<td>22.18</td>
<td>43.56</td>
</tr>
<tr>
<td>arXiv Filtered</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>46.23</td>
<td>18.02</td>
<td>41.02</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>46.36</td>
<td>18.29</td>
<td>41.23</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>46.56</td>
<td>18.27</td>
<td>41.34</td>
</tr>
</tbody>
</table>

Table 3: Result of our models and the compared backbone model (PEGASUS) on the filtered datasets. ROUGE score and Entity(Summ) are mainly used to measure the word-level saliency and entity-level saliency, respectively. Entity(Doc) is used to measure the entity-level factual consistency. Red represents the lowest among all the three models, while Green represents the highest.

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE</th>
<th>Entity(Summ)</th>
<th>Entity(Doc)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-2</td>
<td>R-L</td>
</tr>
<tr>
<td>CNNDM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>44.62</td>
<td>20.82</td>
<td>31.05</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>44.19</td>
<td>20.86</td>
<td>31.19</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>44.16</td>
<td>20.61</td>
<td>30.97</td>
</tr>
<tr>
<td>XSum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>46.65</td>
<td>23.47</td>
<td>38.67</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>46.23</td>
<td>22.76</td>
<td>37.96</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>46.02</td>
<td>22.36</td>
<td>37.58</td>
</tr>
<tr>
<td>Pubmed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>46.11</td>
<td>19.43</td>
<td>41.22</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>46.21</td>
<td>19.86</td>
<td>41.51</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>46.27</td>
<td>19.82</td>
<td>41.59</td>
</tr>
<tr>
<td>arXiv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEGASUS</td>
<td>44.23</td>
<td>16.55</td>
<td>39.15</td>
</tr>
<tr>
<td>SpanCopy</td>
<td>44.05</td>
<td>16.76</td>
<td>38.91</td>
</tr>
<tr>
<td>SpanCopy+GR</td>
<td>44.00</td>
<td>16.87</td>
<td>38.92</td>
</tr>
</tbody>
</table>

Table 4: Result of our models and the compared backbone model (PEGASUS) on the unfiltered datasets. See Table 3 for the details of the columns.

$src_p$ is 100. Comparing SpanCopy and PEGASUS, SpanCopy performs better than PEGASUS regarding both saliency and entity-level factual consistency. Plausibly, this is because all the entities in the ground-truth summary can be copied from the source document, in which case the SpanCopy mechanism can better learn to copy. The SpanCopy model with the GR component performs better regarding the entity-level saliency on three out of all the four datasets. On arXiv, the performance of SpanCopy with the GR component regarding both entity-level saliency and factual consistency is quite low. One likely reason might be that it is a rather difficult task to identify the salient entities in the arxiv dataset, as there is a large amount of entities in the source documents, but only very
As shown in Table 4, the SpanCopy model improves the factual consistency of generated summaries compared with the PEGASUS model on all the datasets. Comparing across the four datasets, SpanCopy outperforms PEGASUS on both the saliency and factual consistency metrics on the Pubmed dataset. For better comparison, we show the relative gains/loss regarding PEGASUS on all the datasets, as well as the overall average results in Table 5. It is clear that the SpanCopy model performs much better regarding entity-level factual consistency (+2.13) with essentially no change in saliency (−0.06 on average ROUGE and −0.04 on entity-level saliency). Admittedly, despite the success of the GR component on the filtered datasets on both word-level and entity-level saliency, it fails to deliver any gain on the original datasets. A plausible explanation is that GR makes the model focus excessively on the entities in the source document, therefore penalizing generation of new, potentially summary-worthy, entities.

Comparing the entity-level factual consistency on the filtered datasets and the original datasets, the filtered datasets always have higher $src_p$ than the original ones, and the gain is especially larger on the XSum and arXiv datasets, as both of them contain more entity-level hallucinations in the original datasets. Remarkably, the performance gain of the SpanCopy model over PEGASUS on the filtered XSum dataset is much larger on the original XSum datasets (7.98 v.s. 0.66), which might be because original XSum is more abstractive, the entity-level guidance is especially helpful for the abstractive examples with consistent entities in the summary.

### 4.5 Qualitative Analysis

For illustration, we examine a real example from the CNNDM dataset in Table 6, which is a news article on the evacuation of Americans during the time of the crossfire of warring parties in Yemen. While all of the three system generated summaries are able to capture the main statement that ‘it’s too dangerous to evacuate the Americans’, the person ‘Ivan Watson’ mentioned by PEGASUS’s summary does not exist in the source document, i.e., it is an ‘hallucinated’ entity. Most likely, PEGASUS is generating such hallucination because ‘Ivan Watson’ is a senior CNN correspondent several time associated with Yemen in other news article in the training set, and the model automatically ‘picked the entity from the memory’ to generate the summary without tightly adhering to the given document. In contrast, both of our models do not contain entities that are not in the source document, as the SpanCopy mechanism tend to guide the model to use more the entities in the source document. In addition, with the GR component, although the generated summary contains more matched entities with the source document, it pushes the model too far towards copying entities which are not salient (e.g. The State Department).

### 5 Conclusion and Future Work

In this paper, we tackle the problem of entity-level factual consistency for abstractive summarization through a discourse-aware approach, by guiding the model to directly copy the summary-worthy entities from the source document, through the novel SpanCopy mechanism (with the optional GR component). This mechanism can be integrated into
Table 6: Example of the entity-level factual inconsistency, taken from the CNNDM dataset. The first block shows the entities in the source document with high GR scores (shown in parenthesis) from the SpanCopy + GR model.

| Entities in the Source Document: | Yemen(0.28), Americans(0.25), Saudi Arabia(0.23), the State Department(0.23), CNN(0.20), U.S.(0.15), ...
| Ground-truth Summary: | No official way out for Americans stranded amid fighting in Yemen. U.S. Deputy Chief of Mission says situation is very dangerous so no mass evacuation is planned.
| PEGASUS: | CNN’s Ivan Watson joins a mother and her grandchildren waiting to be evacuated from Yemen. The State Department has said it is too risky to evacuate Americans from the area. Watson meets Americans who were on a CNN ship that docked at a Yemeni port.
| SpanCopy: | Dozens of Americans are trapped in Yemen. The U.S. has said it is too dangerous to evacuate Americans.
| SpanCopy+GR: | The U.S. has said it is too dangerous to evacuate Americans from Yemen. The State Department said it is too risky to conduct an evacuation of citizens. A group of U.S. organizations have filed a lawsuit against the government’s stance on evacuations.

In addition, we use PEGASUS (Zhang et al., 2020) as our base model in all the experiments on different datasets, as it has delivered top performance on multiple summarization datasets. We follow the original paper on the length limits of all the datasets, however, the length of the source documents in both scientific paper datasets are much longer than the length limit (3k/6k v.s. 1024), which leaves the room for further improvement with sparse attention techniques applied (Xiao et al., 2022; Guo et al., 2022).

Ethics Consideration

Although we tackle the problem of factual inconsistency for abstractive summarization, and improve the entity-level factual consistency of the generated summaries by applying the entity-level span copy mechanism, the generated summaries still contain unfactual information. Therefore, caution must be exercised when the model is deployed in practical settings.

References

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Inspecting the factuality of hallucinated entities in abstractive summarization. CoRR, abs/2109.09784.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).


Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. 2022. Don’t say what you don’t know: Improving the consistency of abstractive summarization by constraining beam search.


A Model and Training Details

We use PEGASUS as our backbone model, which contains 571M parameters, and the span copy mechanism has 2M additional parameters. We train the fine-tuned models from the huggingface model hub\(^6\) for 100k steps (16 data per step) with early stopping based on the ROUGE scores on the validation set, which takes around 24 hours with single V100 GPU.

B Datasets

We compare the size of filtered and original datasets in Table 7.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Data (original)</th>
<th># Data (filtered)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNNDM</td>
<td>287,113/13,368/13,368</td>
<td>105,847/4,490/3,903</td>
</tr>
<tr>
<td>XSum</td>
<td>204,017/11,327/11,333</td>
<td>42,481/2,349/2,412</td>
</tr>
<tr>
<td>Pubmed</td>
<td>119,924/6,633/6,658</td>
<td>32,123/1,797/1,772</td>
</tr>
<tr>
<td>arXiv</td>
<td>202,914/6,436/6,440</td>
<td>66,360/2,365/2,324</td>
</tr>
</tbody>
</table>

Table 7: Number of data examples in all the datasets (original v.s. filtered).

C Software and Licenses

Our code is licensed under Apache License 2.0. Our framework dependencies are:

- HuggingFace Datasets\(^7\), Apache 2.0
- NLTK \(^8\), Apache 2.0
- Numpy\(^9\), BSD 3-Clause "New" or "Revised"
- Spacy\(^10\), MIT
- Transformers\(^11\), Apache 2.0
- Pytorch\(^12\), Misc
- Pytorch Lightning\(^13\), Apache 2.0
- ROUGE\(^14\), Apache 2.0

\(^6\)https://huggingface.co/models
\(^7\)https://github.com/huggingface/datasets/blob/master/LICENSE
\(^8\)https://github.com/nltk/nltk
\(^9\)https://github.com/numpy/numpy/blob/main/LICENSE.txt
\(^10\)https://github.com/explosion/spaCy/blob/master/LICENSE
\(^11\)https://github.com/huggingface/transformers/blob/master/LICENSE
\(^12\)https://github.com/pytorch/pytorch/blob/master/LICENSE
\(^13\)https://github.com/PyTorchLightning/pytorch-lightning/blob/master/LICENSE
\(^14\)https://github.com/google-research/google-research/tree/master/rouge
Discourse Information for Document-Level Temporal Dependency Parsing

Jingcheng Niu\textsuperscript{123}, Victoria Ng\textsuperscript{2}, Erin E. Rees\textsuperscript{2}, Simon de Montigny\textsuperscript{24}, Gerald Penn\textsuperscript{13}

\texttt{\{niu,gpenn\}@cs.toronto.edu} \texttt{\{victoria.ng,erin.rees\}@phac-aspc.gc.ca} \texttt{simon.de.montigny@umontreal.ca}

University of Toronto\textsuperscript{1}, Public Health Agency of Canada\textsuperscript{2}, Vector Institute\textsuperscript{3}, University of Montreal\textsuperscript{4}

Abstract

In this study, we examine the benefits of incorporating discourse information into document-level temporal dependency parsing. Specifically, we evaluate the effectiveness of integrating both high-level discourse profiling information, which describes the discourse function of sentences, and surface-level sentence position information into temporal dependency graph (TDG) parsing. Unexpectedly, our results suggest that simple sentence position information, particularly when encoded using our novel sentence-position embedding method, performs the best, perhaps because it does not rely on noisy model-generated feature inputs. Our proposed system surpasses the current state-of-the-art TDG parsing systems in performance.

Furthermore, we aim to broaden the discussion on the relationship between temporal dependency parsing and discourse analysis, given the substantial similarities shared between the two tasks. We argue that discourse analysis results should not be merely regarded as an additional input feature for temporal dependency parsing. Instead, adopting advanced discourse analysis techniques and research insights can lead to more effective and comprehensive approaches to temporal information extraction tasks.

1 Introduction

Temporal Information Extraction (TIE) is the task of modelling the relative and/or absolute temporal relations between all the temporal nodes in an article. A temporal node can be either an event or a time expression (timex). TIE is a core component task of text comprehension. Despite its importance, TIE remains one of the lowest performing natural language understanding tasks. It is a difficult task, and the challenge is further compounded when expanding it to the document level, as the number of temporal relations scales quadratically with the number of temporal nodes, and the required amount of reasoning must incorporate longer spans of text.

To address these challenges, Kolomiyets et al. (2012); Zhang and Xue (2018b); Yao et al. (2020) proposed the use of temporal dependency structures to represent the overall temporal relational structure within an article. This approach is based on the phenomenon of temporal anaphora, where the interpretation of the occurring time of one temporal node depends on knowing the occurring time of another temporal node. By modelling these temporal dependency relations, the overall temporal structure of an article can be obtained without the need for exhaustively labelling every pair of temporal nodes.

As a result, temporal dependency parsing not only models the temporal relations between events but also captures narrative and discourse structure. There are striking similarities between temporal dependency structures and the constituency discourse tree structures (Guz and Carenini, 2020) used for discourse parsing in the context of Rhetorical Structure Theory (RST; Mann and Thompson, 1988), and not just in their use of trees or graphs. More importantly, temporal dependency relations can be viewed as a specific type of anaphoric relation that discourse analysis models attempt to capture. This observation suggests a potential connection between dependency parsing and discourse analysis, warranting further investigation into their relationship and potential synergies.

This connection between document-level temporal structure and discourse structure was corroborated by Choubey and Huang (2022), who discovered that incorporating discourse profiling (DP) information, specifically the functional role of each sentence, could enhance the overall performance of temporal dependency graph parsing (TDG; Yao et al., 2020). Their evaluation may not have been sufficiently comprehensive, however. TDG parsing encompasses three distinct types of relation parsing:
timex to timex (t2t), event to timex (e2t), and event to event (e2e), each requiring a different prediction mechanism. Upon a more detailed reexamination of Choubey and Huang’s (2022) findings, DP information in fact does not consistently improve performance across all three relation types; it reliably enhances e2e, but may lead to a decline in performance for the other two.

We believe this is caused by two major limitations of Choubey and Huang’s (2022) approach. First, DP is a hard problem in its own right. The state-of-the-art DP system (Choubey and Huang, 2021) only yields a 59.21% F1 performance. This means TIE systems following Choubey and Huang’s (2022) guidance will only have access to noisy and inaccurate DP features. Second, sentence function is a relatively high-level, descriptive type of discourse structure. Temporal dependency structure, on the other hand, can also benefit from a lot of simple surface-level discourse information, such as precedence (Zhang and Xue, 2018a).

To address these issues, we have experimented with incorporating surface-level sentence-position information into a TIE system, and in two ways: encoding absolute sentence-position by appending the sentence number directly onto the context sentences, following Choubey and Huang (2022), and proposing a novel Sentence Position Embedding (SPE) using a sinusoid. Our experiments demonstrate that SPE could significantly enhance temporal dependency graph parsing performance across all relation types, with the performance increase being mostly greater or at least comparable to that provided by DP information. The resulting TDG parsing system\(^1\) with SPE obtains the state-of-the-art performance.

### 2 Temporal Dependency Parsing

TIE is the task of classifying the temporal relation between two temporal nodes. A temporal node can be either an event trigger (a.k.a. event mention) that represents an event that exists in the narrative of an article, or a timex that is a nominal description of a date or time. When treating a pair of temporal nodes as either intervals or points on the timeline, the temporal relation between temporal nodes can be described by Allen’s (1983) temporal calculus. There are some variations between different TIE annotation standards, but generally speaking, temporal relations include links such as BEFORE, AFTER and OVERLAP.

This pairwise annotation scheme, however, fails to generalize to the document level. The number of temporal node pairs is quadratic in the number of temporal nodes (\(n\), i.e., \(\binom{n}{2}\) \(\in O(n^2)\). Yao et al. (2020) pointed out that this quadratic increase, together with the increase in the complexity and number of vague relation links for annotators to consider will, in practice, inevitably cause errors to annotation.

To address this issue, Kolomiyets et al. (2012); Zhang and Xue (2018b); Yao et al. (2020) have advocated for using dependency structures to represent document-level temporal relations. Kolomiyets et al. (2012) annotated a children’s story with temporal dependency trees. Each event \(u\) only depends on one other event \(v\) iff the interpretation of when \(u\) occurred requires knowing when \(v\) occurred. Kolomiyets et al.’s (2012) temporal dependency tree structure only includes events, but this standard may yield disconnected structures. Zhang and Xue (2018b) refined temporal dependency tree structure to allow the inclusion of timex

---

\(^1\)The code and data are publicly available online: https://github.com/frankniujc/tdg-discourse.
vertices as well as two special vertices: a document creation time (DCT) vertex and a ROOT vertex. The inclusion of timex vertices allows for capturing the missing events in timex (e2t) temporal dependencies and timex to timex (t2t) temporal dependencies. The addition of the DCT and ROOT vertices ensures each document is always parsed into a valid TDT.

Both Kolomiyets et al. (2012) and Zhang and Xue (2018b) assumed that each event or timex had exactly one reference temporal node (to which the dependency edge points), resulting in a tree structure. Yao et al. (2020), on the other hand, argued that this assumption is overly stringent, and that it is possible for an event to have both a reference timex and an reference event. They therefore proposed to characterise temporal structure with temporal dependency graphs (TDG), in which each event can have a timex parent, an event parent, or both. As depicted in Figure 1, the event alerted depends on both the timex Friday and the event fall. As a result, TDG is more expressive than the earlier TDTs. In this work, we used the TDG corpus released by Yao et al. (2020). Table 1 shows the statistics of this corpus.²

3 Model Architectures

3.1 Pairwise Classification Model

Typically, TIE is formulated as a classification task. Given a pair of temporal nodes (n₁, n₂), the sentences containing the nodes ([t₁₁, . . . , t₁m], [t₂₁, . . . , t₂n]) are encoded into a context vector e = [e₁₁, . . . , e₁m, e₂₁, . . . , e₂n]. Next, the event embedding pair [eₙ₁; eₙ₂] are concatenated and the classification task is performed using a multilayer perceptron (MLP) layer. Where a temporal node spans multiple tokens, we utilize Lee et al.’s (2017) method for obtaining an attentive span representation. Figure 2 depicts an overview of this architecture. In this model, we deliberately avoid jointly learning the pairwise model to observe the effects of different discourse information on various relation types.

3.2 Joint Ranking Model

Neural ranking models (Zhang and Xue, 2018a; Ross et al., 2020; Choubey and Huang, 2022) formulate the task as a regression problem. For each temporal node, the model predicts a scalar reference score for every potential parent node and selects the edge with the highest reference score. Therefore, this edge selection process can be formulated as a classification task — the reference scores can be considered as classification logits, and the cross-entropy loss of the edge prediction can be calculated. The three relation types (t2t, e2t, and e2e) are trained jointly. Unlike the pairwise model that uses the concatenation of the two event embeddings, we follow Choubey and Huang (2022), who enclose both triggers in special symbols ($n₁$ and #n₂#) and use the embedding of the [CLS] token as the pair embedding $e_{n₁,n₂} = e_{[CLS]}$.

²There are some minor discrepancies between the statistics reported by Yao et al. (2020) and the final released corpus. We used the final version of the TDG corpus released at https://github.com/Jryao/temporal_dependency_graphs_crowdsourcing.
4 Discourse Analysis for TDG Parsing

Based on Dijk’s (1986) schemata of news content, Choubey et al. (2020) proposed the task of discourse profiling (DP). The task is to classify each sentence into one of eight content types (see appendix B). There are two ways of encoding DP information, as proposed by Choubey and Huang (2022). The first (DP Feature) appends the content type label directly, marked with a special token $. For instance, if the sentence represents a main event, the label $\#M1\#$ is appended to the sentence. We obtained the same model-generated content type labels from Choubey and Huang (2022). The second (DP Distillation) involves using model distillation. In this approach, the model is equipped with two decoders: one predicts the reference score, while the other performs DP classification. The training of both tasks occurs simultaneously, distilling the DP information into the underlying language model.

4.1 Sentence Position Information

Sentence position information has proven valuable in various tasks. For instance, the next sentence prediction (NSP) task played a crucial role in training BERT (Devlin et al., 2019), and similar techniques have been shown to be effective for discourse analysis (Yu et al., 2022). In temporal dependency parsing, previous work (Zhang and Xue, 2018a) employed hand-crafted precedence features to enhance performance. In this study, we also present two methods for encoding sentence position:

Sentence Position Feature (SPF) We experimented with directly incorporating sentence position information into the context sentence, in a manner similar to the DP feature. For each sentence, we prepend the context sentences with “Sentence X:,” where X represents the sentence number.

Sentence Position Embedding Vaswani et al. (2017) utilized sine and cosine functions with varying frequencies for token position encoding. We extend this idea by proposing a sentence position encoding (SPE; Equation 1), where pos denotes the sentence number, i is the dimension, and $d_{model}$ is the model’s dimension.

$$SPE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$
$$SPE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (1)$$

Since the SPE shares the same dimension as RoBERTa’s word embeddings, they can be summed. For the pairwise model, we add the SPE of the event’s sentence to its event embedding. For the joint ranking model, we directly add both SPEs of both sentences to the pooler’s output. A post-hoc classifier on RoBERTa itself serves as our baseline.

5 Experimental Results

5.1 Pairwise Prediction Results

The left side of Table 2 displays the performance of the models with various types of discourse information. Among the results, we can emphasize two key comparisons. First, the addition of all kinds of discourse information leads to a substantial performance increase in the e2e parsing task; however, it may result in a decline in performance for the other two types. A contributing factor is that the e2e task not only models temporal dependency structure but also requires the model to learn a shortcut heuristic that takes sequence length into account. Upon closer examination, we discovered that Yao et al.’s (2020) assumption that each event can depend on at most one other event is not always valid. It is common for an event to have multiple parents. In such cases, the TDG annotation standard instructs the annotator to choose the event that is closer in time. If this is not feasible, the annotator should select the event that is closer in textual order. Therefore, discourse information offers extra benefits for e2e parsing, regardless of the DP encoding.

Second, SPE is the only information that leads to performance improvements across all three relation types, and it also yields the most significant performance increase. As previously discussed, DP information that is model-generated is noisy. Moreover, the discourse structure of TDG news articles is relatively simple. Surface-level sentence position can be considered a reliable proxy for the article’s discourse structure. For instance, every news article in the TDG corpus begins with a timex indicating the publication date of the article. Additionally, the majority of the articles follow the publication time with the lead sentence of the article. Directly incorporating the sentence number into the article, however, does not produce the same level of performance improvement. This outcome is also expected, as a BERT-based language model struggles with representing numbers (Wallace et al., 2019).
5.2 Ranking Model Results

The right side of Table 2 presents the performance of the ranking models. Once again, SPE achieves the highest overall performance, showcasing the effectiveness of this approach. Similar to the pairwise results, all models surpass the baseline for the e2e task. Interestingly, with only a few exceptions, the e2t and t2t performance of each model declines. In addition to the previously mentioned reasons, one contributing factor is the imbalanced distribution of the three relation types. The TDG corpus contains 2,486 timexes and 14,779 events, resulting in 20,862 t2t, 63,065 e2t, and 233,065 e2e potential pairs in the training set. When all three types are trained jointly, the model overfits on the e2t and e2e relations, leading to performance disparities across the three relation types.

Despite the issue of data imbalance, the benefits of joint learning are substantial. All models exhibit better performance on the e2t and e2e tasks compared to their pairwise counterparts. The three relation types are not disconnected; for instance, events that depend on the same timex are likely to depend on each other. Without joint learning, this valuable TDG structural information is lost. There are moreover several ways to better model structural information, such as the application of GNNs (Ji et al., 2019), as well as methods to address the data imbalance issue. We leave these topics for future research.

6 Discussion

Before Choubey and Huang (2022), the relationship between discourse and TIE had not been explored, and indeed our own experiments corroborate the value of their insight to incorporate discourse information into constructing document-level temporal structures. Merely using the output of a discourse system as an additional input feature for document-level TIE may not be the most effective strategy, however. A very superficial, but novel sentence position embedding effectively encodes surface-level sentence-order information, and seems to be more reliable as a proxy for the discourse structure of news articles. Incorporating this information leads to state-of-the-art performance in TDG parsing.

The success of sentence-position embedding offers a significant opportunity to bridge discourse analysis and document-level temporal dependency parsing. It suggests that we should not naively rely on discourse information as a separate, modular input source. Instead, the similarities between the two tasks indicate that various techniques and insights can be transferred and applied across both domains, leading to more effective models and a deeper understanding of the relationship between discourse analysis and temporal dependency parsing.

Acknowledgement

This study is funded by the Canadian Safety and Security Program (CSSP) from Defence Research and Development Canada (DRDC) awarded to the Public Health Agency of Canada (CSSP-2018-CP-2334: Incorporating Advanced Data Analytics into a Health Intelligence Surveillance System). We thank the Global Public Health Intelligence Network (GPHIN) and Epidemic Intelligence from Open Sources (EIOS) teams for their support. We would also like to thank Prafulla Kumar Choubey for sharing the DP data.

Limitations

In accord with Choubey and Huang (2022), our study focuses solely on the unlabelled performance of TDG parsing. This implies that our evaluation is limited to identifying reference temporal relations without considering the classification of relation types. We plan to explore the labelled TDG parsing task in future research.
Owing to resource constraints, our experiments were conducted using only one type of language model, RoBERTa-base. However, other models such as BERT (Devlin et al., 2019), DeBERTa (He et al., 2021), and ERNIE (Zhang et al., 2019) have demonstrated impressive performance across various natural language understanding benchmarks. We aim to evaluate these models in future research, and we encourage other researchers to reproduce our work using these alternative models.

References


A Training Details

We follow Choubey and Huang’s (2022) experiment setup. We first conducted a hyperparameter search on learning rate using the baseline models. In particular, we used 1e-5 for the t2t pairwise models, 3e-5 for the e2t and e2e pairwise models, and 8e-5 for the joint ranking models. We train each model for 15 epochs, and report the test set performance on the model with the highest development set performance. RoBERTa-base is used as the encoder for all the experiments. For the pairwise model, we down sampled e2e labels by a factor of 10.

B DP Content Types

Choubey et al. (2020) specified eight DP content types: Main event (M1), Consequence (M2), Previous Event (C1), Current Context (C2), Historical Event (D1), Anecdotal Event (D2), Evaluation (D3) and Expectation (D4).
Encoding Discourse Structure: Comparison of RST and QUD

Sara Shahmohammadi¹*, Hannah Seemann²*, Manfred Stede¹, Tatjana Scheffler²

¹ Department of Linguistics, University of Potsdam, Germany
² Department for German Language and Literature, Ruhr-University Bochum, Germany

shahmohammadi@uni-potsdam.de  hannah.seemann@rub.de
stede@uni-potsdam.de  tatjana.scheffler@rub.de

Abstract

We present a quantitative and qualitative comparison of the discourse trees defined by the Rhetorical Structure Theory and Questions under Discussion models. Based on an empirical analysis of parallel annotations for 28 texts (blog posts and podcast transcripts), we conclude that both discourse frameworks capture similar structural information. The qualitative analysis shows that while complex discourse units often match between analyses, QUD structures do not indicate the centrality of segments.

1 Introduction

Rhetorical structure Theory (RST) (Mann and Thompson, 1988) and the Question under Discussion (QUD) model (e.g., Ginzburg, 1996; Roberts, 2012; Onea, 2019) are two accounts of discourse structure that stem from different research fields and aim to explain different phenomena (speaker intentions and rhetoric versus information structure). However, they share a fundamental formal assumption: that discourse structure is to be represented as a tree that is constructed by recursively combining adjacent “elementary units” of the discourse. For QUD, “discourse” originally meant primarily dialog, while RST was designed for monologue text. Nonetheless, researchers have occasionally explored ways to apply one theory also to the mode of the other.

In this paper, we systematically compare these two approaches to discourse structure, based on empirical observations in a novel multi-media corpus. While some researchers have previously noted the intuitive similarities of RST based trees and QUD based trees on a theoretical level, this work presents the first study where both frameworks are systematically applied to a corpus of both spoken and written data, and compared in a quantitative and qualitative manner.

*Equal contribution

We first present part of a novel corpus of German blog posts (monologue) and podcast transcripts (dialog). There is a loose 1:1 correspondence between the two, in that the blogs are descriptions of what is being discussed in the podcasts. To our knowledge, this is one of the first corpora that are annotated in parallel with RST and QUD structures. Our aim is to compare the annotated material so that insights into the descriptive and explanatory power of the two approaches can be gained from an empirical perspective of studying authentic data. We make the annotated corpus available to facilitate follow-up research.

To enable the quantitative comparison, we automatically map manually-annotated RST trees to Riester (2019)-style QUD trees. We make the conversion tool available as a web application (and will later release the code). Based on the common format, we perform a quantitative analysis of the similarity of RST and QUD discourse trees, for which we propose an evaluation measure. In addition, our thorough qualitative comparisons show that QUD trees do not indicate the centrality of segments, and often fail to cover relations such as concession and contrast. On the other hand, the topic progression and speaker change within a dialog is captured in QUD analyses but may be missing from RST.

2 Related Work

In annotating both media present in our corpus, we apply both discourse models outside of what they have been primarily designed for. While QUD has been applied to monologue texts before (Riester et al., 2018; Westera et al., 2020), it is most centrally applied to (short) dialogs. On the other hand, RST aims to capture the intentional structure constructed by the writer and is thus designed for monologue, but it has been occasionally applied to dialog, as well (e.g., Stent, 2000).

The QUD framework has been developed to capture aspects of the information structure of sen-
ences and certain specific pragmatic phenomena (Ginzburg, 1996; Roberts, 2012). Only recently has it been used for annotating larger texts, and compared to models of the coherence structure of discourse.

Hunter and Abrusán (2017) compare QUD structures to those proposed by Segmented Discourse Representation Theory (SDRT). They argue that although there likely is no QUD corresponding to every discourse relation, “QUDs correspond to complex discourse units in a discourse graph” (p. 41), that is, topics that lead to grouping discourse units together.

Onea (2019) similarly starts by comparing QUD and SDRT and argues that formally analyzing the erotetic (i.e., question) structure of a discourse can be useful to understand its meaning and its relation structure, for example its SDRT representation. He develops a method for mapping (parts of) question graphs to SDRT representations, and takes a close look at the Result relation as a case study. He argues that QUD theories (in particular, models based on potential questions) have repercussions for the larger discourse structure of a text (as represented for example in SDRT).

Riester et al. (2018) offer the first detailed guidelines for segmenting discourse and annotating QUD trees in authentic text, discussing individual texts from English, French, and German and from three different genres. One claim is that the same guidelines apply to monologue (newspaper articles) and dialog (interviews) alike. In later work, Riester et al. (2021) compare QUD, RST, and the CCR approach to discourse structure for one text. Regarding segmentation rules in QUD and RST, they point out that QUD calls for smaller segments than RST when information-structural factors suggest a discourse contribution, e.g., for contrastive foci. Conversely, adjunct clauses can sometimes be separate segments in RST (e.g., in Circumstance relations) but not in QUD. As for the relations, the authors show how RST relations can be integrated into a QUD tree notation, and discuss some typical mappings.

Riester (2019) presents a proposal to include both sub- and coordinating relations in a tree that combines QUD and SDRT, based on the approach by Klein and von Stutterheim (1987). For example, the temporal progression of a discourse can be represented by questions asking about each point in time: What happened at $t_{1:n}$?

Finally, we mention that an early (but somewhat inconclusive) debate in the RST community on the interplay of “intentional” and “informational” coherence relations (e.g., (Moore and Pollack, 1992)) foreshadowed the kind of duality that RST and QUD embody.

3 Data and Method

We carry out an empirical comparison of QUD- and RST-based annotations of the same texts in two media. The idea is based on the assumption that while the QUD and RST frameworks cannot be directly mapped onto one another, both aim to capture the overall coherence of a discourse in a tree-like fashion. Thus, previous work such as (Hunter and Abrusán, 2017; Riester et al., 2021) has proposed to study the correspondences in these discourse trees by looking at the relation between rhetorical relations and question-based structures. Our study is the first, to our knowledge, which carries out a parallel annotation of both spoken and written texts in the two frameworks.

Our data and annotation process is described here.

3.1 Data

The corpus contains texts from two media: podcast transcripts and their corresponding blog posts, both in German. Furthermore, the corpus contains different domains: business podcasts that are produced by companies like DELL or Deutsche Telekom and science podcasts that cover topics from various fields of science and politics. For this analysis, we use 14 blog posts and chunks of 14 podcast transcripts. The blog posts are composed of 26 EDUs on average. The contiguous discourse chunks we annotated from the transcripts consist of 17 EDUs on average. Table 1 shows the size of the resulting sub-corpus.

<table>
<thead>
<tr>
<th>medium</th>
<th># episodes</th>
<th># EDUs</th>
<th># tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>blog posts</td>
<td>14</td>
<td>364</td>
<td>4,204</td>
</tr>
<tr>
<td>transcripts</td>
<td>14</td>
<td>502</td>
<td>4,980</td>
</tr>
<tr>
<td>total</td>
<td>28</td>
<td>866</td>
<td>9,184</td>
</tr>
</tbody>
</table>

Table 1: Corpus overview, EDU and token count.

3.2 Annotation

The texts have been manually annotated in both frameworks, RST and QUD. To simplify the com-
parison between the annotations, we use the RST segmentation according to the Potsdam Commentary Corpus guidelines (Stede, 2016) for both frameworks. For now, we assume EDUs as viable segments for QUD annotation, even though there are differences compared to the usual QUD segmentation, as discussed by Riester et al. (2021). Other than the segmentation, the QUD annotation follows the guidelines defined by Riester et al. (2018). The RST annotation mainly follows the guidelines proposed by Stede et al. (2017), with a few changes.¹

The annotations were conducted by one person for each model, and revised by a second annotator (a co-author of this paper).² The annotated files can be found in the project’s GitHub repository³. For reasons of space, not all examples referred to in our analysis are shown in the text, but they can be found under the file name given, e.g. CRE210_Transcript_p3. An example of the same file annotated with both discourse models is presented in Figure 2. The English translation of this example is given in (4).⁴

4 Converting RST to QUD Trees

Both RST structures and QUD trees encode discourse structure formally as trees that span over the entire discourse. In RST, intermediate nodes are discourse relations that group (typically) two segments, (typically) a nucleus and satellite. In QUD, intermediate nodes are explicit or implicit questions which guide the discourse; children are (partial) answers to these questions. Disregarding node labels for intermediate nodes, trees with the same yield can be mapped onto each other by comparing just the branching structure.

To evaluate the similarity of RST and QUD trees, we converted the RST trees to a format similar to the QUD trees that can be quantitatively compared to the QUD annotation. Figure 1 shows the converted version of the RST tree in Figure 2.

To convert an RST tree to the QUD format, we take a discourse relation in an RST tree as an intermediate node (implicit question) in a QUD structure. The satellite and nucleus of the relation are daughters of this intermediate node at the same level of nesting.⁵ The details of the conversion will be made available in the project repository on GitHub.

5 Quantitative Analysis of RST and QUD Correspondence

We automatically evaluated the similarity of the RST and QUD discourse structures quantitatively.

¹To account for particularities of speech, we added a ‘completion’ relation that is used in podcast conversations, if a speaker says something that is not complete, e.g. does not have a verb, and then completes it later. In addition, we extended the ‘restatement’ relation to allow being used as a forward-looking relation. This way, it also covers cases of a speaker’s self-correction. It is noteworthy to know that there is no ‘question’ relation in these guidelines. Such a relation exists in some RST guidelines, for instance in the annotation guide proposed by Carlson and Marcu (2001).

²For future work, we will add a second annotation in each framework and inter-annotator-agreement.

³https://github.com/mohamadi-sara20/rst-qud-comparison

⁴All examples are our own translations of the original German data.

⁵It is also possible to convert an RST tree into a QUD tree where the satellite is nested one more level compared to the nucleus. That is, to consider the satellite a subtopic of the nucleus. This way, information on nuclearity will not be lost in the conversion. However, we found this less similar to a QUD structure, so it was not used for the final analysis.
[Q What is the way things are?]
- Q: Welche Folge hat das?
- A: Für alle anderen Meldungen, die davor reingekommen ist, musste man den Datendump vom Vortag kennen.
- Q2: Ist das möglich?
- A: Es gibt aber kein vorangehendes Archiv.
- Q3: Gibt es andere Möglichkeiten als ein öffentliches Archiv?
  - und weil ich erst von der Woche damit angefangen habe,
  - habe ich jetzt auch erst auch dank noch jemandem auf Twitter so, der mir noch zwei fehlende Dumps, dass ich jetzt eine woche habe an an Dumps
- Q4: Wie bewertet S1 diese Situation?
  - S1: Es gibt keinen.
  - das macht mich jetzt noch ein bisschen fassungslos.
- Q5: Wie genau ist die Situation?
  - Die Daten sind immer nur tagesaktuell.
  - und eröffnen sich dann der laufenden Meter.
  - und es gibt kein Archiv der letzten Daten.
  - Q6: Warum ist ein Archiv notwendig?
  - obwohl man genau die brachte.
  - und die volle Information herauslesen zu können.
- Q7: Bestätigt S2 diese Zusammenfassung?
  - S2: So ist es so.
- Q8: Wie bewertet S1 die Situation?
  - S1: Manchmal geht es gott.

Figure 2: Representations of (4) in QUD (top/left) and RST (bottom/right) (UKW024-p3).
across the whole corpus. To do this, we computed a variant of the PARSEVAL measure known from evaluating (syntactic) constituency parse trees. This measure compares unlabelled trees A and B: First, for all nodes \( N_A \) in tree A, one determines whether tree B contains a node with the same yield (= concatenation of all the text dominated by a node) as \( N_A \). In the case of comparing an automatic parse tree with a gold standard, this would reflect precision. Second, for all nodes \( N_B \) in B, the same is repeated (~ recall). We finally compute the harmonic mean of both directions to determine the similarity of RST and QUD structures.

We use unlabelled parseval scores because intermediate nodes are labelled with relations in the case of RST trees, and with questions in QUD trees.\(^6\) Note that the standard parseval score includes leaf nodes in its computation (in the case of syntactic trees, POS tags), and that this practice typically leads to much higher scores than excluding leaves. In our computation, we also include leaves, not only because it is commonly done, but also because in our tree structures, certain error cases can only be reflected when including leaf nodes. This happens in particular, when an explicit text segment is used as an explicit question under discussion (i.e., internal node) in a QUD tree. For this reason, we also redefine the “yield” of a node to be all explicit text dominated by that node, both when it is represented in a leaf node and when it is in an intermediate node.

<table>
<thead>
<tr>
<th>medium</th>
<th>Q→R</th>
<th>R→Q</th>
<th>f-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>blogs</td>
<td>0.87</td>
<td>0.68</td>
<td>0.76</td>
</tr>
<tr>
<td>transcripts</td>
<td>0.85</td>
<td>0.63</td>
<td>0.73</td>
</tr>
<tr>
<td>total</td>
<td>0.86</td>
<td>0.65</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table 2: (Micro-averaged) parseval scores comparing RST and QUD discourse trees.

The results of the quantitative comparison are shown in Table 2. It can be seen that there is a large amount of overlap in the tree structures between QUD and RST frameworks, with an average similarity (parseval) score of 0.74. In addition, we see that the blog posts show higher similarity across frameworks than the podcast transcripts. This is the case even though the blog posts annotated here are on average longer than the transcript snippets (in a longer discourse, there are more possibilities for mismatches in discourse structure annotations). However, we can observe that the similarity of discourse structure between QUD and RST trees is quite high, comparable to inter-annotator agreement within the same framework.

6 Qualitative Comparison of RST and QUD Trees

To further evaluate the correspondences between the analyses, we take a closer look at our annotations. First, we inspected the five pairs of trees that received the lowest matching scores in the quantitative comparison; here we note that a frequent source of mismatch is RST’s tendency to build a complex discourse unit in cases where QUD attaches the material locally (see below). Then we turned also to the other pairs, trying to generalize sources of misalignment. Section 6.1 compares the way complex discourse units are constructed in both models. The overall structure of texts annotated with both models seems to often be similar, yet there are instances where the structures are quite different. Sections 6.2 and 6.3 compare how translatable different rhetorical relations are to QUD trees and whether RST trees are able to represent typical characteristics of dialog, like speaker changes.

6.1 Comparison of Complex Discourse Units

As discussed in the previous section, both RST trees and QUD trees seem to similarly cluster EDUs into groups, which is particularly beneficial in higher-level units. It is possible to decompose an RST tree into prominent sub-trees. EDUs in the same sub-tree or cluster are closer to each other than they are to other EDUs. In a QUD tree, EDUs grouped together are put under the same parent question and hence address the same topic or rather, answer sub-questions of an overarching question under discussion. One example can be seen in the RST tree in Figure 2: EDUs 1–6 make up one cluster, while EDUs 7–14 are grouped together.

The tendency for QUD and RST analyses to group discourse units similarly is noticeable in most of the trees of the corpus, but there are also exceptions. Figure 2 shows an example where QUD
and RST trees seem to capture different aspects of the functions of discourse unit 6. As evident in the figure, RST subtree 1-5 discusses the problem of the unavailability of public data archives. Unit 6 repeats the same idea, without the details. According to the guidelines, one possible relation holding between 6 and 1-5 is the Summary relation. On the other hand, unit 7 starts with the pronoun ‘Das’ (‘that’), which expresses an evaluation of the current condition of the archives. Hence, the best attachment point for it is unit 6.

However, if the evaluation is attached to 6, the summary relation cannot be chosen to relate it to the prior discourse. On the other hand, if the evaluation is attached to 1-6 instead of only 6, it would mean ignoring the intention behind this repetition. This is where the difference between the two trees arises: The RST tree groups 6 with the previous discourse, and therefore loses the ideal attachment point for the evaluation. The QUD tree, on the other hand, groups 6 with the discourse following it, and hence fails to fully observe the function of this repetition.

This pattern is not limited to discourse management in dialogs but is also seen in monologues. In example (1), part (1-a) discusses the fact that there is a great difference between theory and practice in IT security, and brings some evidence to support this claim. Part (1-b) repeats this idea with fewer details and different wording, and (1-c) evaluates the situation.

(1) a. And what we actually see in crime out there is that the so-called cybercrime is working on a level way lower than, for example, academic research. In academic research, we invent amazing new procedures of cryptography, helping us against quantum computers. What I think, I think this research is important. But what really happens to us is that companies are being hacked because some recipients click on email attachments.

b. Meaning, there is a huge difference between the technically complex and artistic attacks in academic research and street crime.

c. And it makes sense if you compare that to a purse thief.

The main reason for adding (1-b) is to bring attention back to the ‘difference between theory and practice in IT security’ in order to later evaluate it. If the evaluation is attached to (1-b), the summary relation cannot be chosen. If the evaluation is attached to Summary(a, b) instead of just (1-b), it would mean ignoring the intention behind this repetition. This is, again, the source of the difference between the two trees: The RST tree groups (1-b) with the previous discourse, and hence loses the ideal attachment point for the evaluation, while the QUD tree groups it with the discourse following it, neglecting the function of the repetition.

Another example of the two representations capturing different aspects of a conversation is shown in (2). In this example, the second sentence by speaker 2 (2-c) has two functions: It is an evaluation of the speaker’s knowledge of the current topic but at the same time part of a turn-taking mechanism, offering speaker 1 to evaluate on the net research.

(2) Context: The first human settlements, beginnings of agriculture, and domestication of animals.

a. Speaker 1: I just looked it up, and it is not really well... It seems to be disputed when the dog really joined us, the process seems to have been over a very long time and gradually.

b. Speaker 2: Yes, maybe it fluctuates, too, that is, how cultures reacted to it.

c. Well, with your net research, you probably know more about it than I do.

d. S1: Well, I’m not reading all of it now. There is a lot to say here about canis lupus, but sometime around 15,000 and 100,000 years. It really is vaguely defined.

(DELL001_Transcript)

Double functions like these are not rare in our data, many utterances have a discourse-managing function on top of the propositional content. Thus, they may be interpreted differently by different annotators, leading to a different annotation of the same text independent of systematic differences between the two discourse models.

6.2 Comparison Between Relations

Restatement. Figure 4 shows an example of a ‘restatement’ relation in both QUD and RST trees.
Instances of ‘restatement’ are all at the same level in a QUD tree (since they all answer the same question), while an RST tree does not represent this as a parallel structure; it forms a tree that is right-branching. Our RST annotation guidelines allow the restatement of adjacent units to be successfully modeled.\(^7\)

Although restating an idea does not introduce a new topic, there are definitely intentions behind it; sometimes the speaker needs time to think and so they buy time by repeating themselves, sometimes interlocutors want to make a topic more memorable and do so by restating it. It is also possible to restate a previous topic, which is not the current topic, in order to make it salient in current discourse. A QUD structure would not be concerned with these functions of a ‘restatement’ relation. In contrast, the fine-grained representation of discourse relations provided in RST or SDRT would distinguish such cases of ‘restatement’.

\(^7\)Our guidelines respect the adjacency constraint, so it is not possible to mark non-adjacent restatements. That is, if somewhere the fifth EDU restates the content of the first EDU, we cannot model that with a restatement relation according to our guidelines, as it would need a non-adjacent edge.

**Background.** A background relation in RST introduces background information in order to enable the reader to understand a more central claim. What is annotated with a ‘background’ relation in an RST tree has been modeled in different fashions in the QUD annotation. Sometimes it is represented in the QUD tree with a nested structure, with a series of nested questions and answers leading to the more central claim. But in different cases what is a ‘background’ satellite in the RST tree is annotated in the QUD tree to share the same immediate parent of the central claim – and therefore without a nested structure.

This is probably due to the QUD model being concerned with different aspects of discourse than the RST model. A QUD representation does not aim at presenting what is the most important or central claim of discourse but rather the series of sub-questions that are talked about.

**Concession and Contrast.** Since there is no question type that can be answered with a ‘but’-statement (Scheffler, 2013), it may be expected that a QUD representation of a discourse is unable to model contrastive relations. Q9 in the QUD tree shown in Figure 3 is annotated with the ‘concession’ relation in the corresponding RST tree.

In the QUD tree, the concession is represented by an additional sub-question – as is also suggested by Riester et al. (2021). This way, concessions can, in principle, be modeled in a QUD representation. However, the QUD representation cannot explicitly show which part of the concession is the expectation or the violation of the expectation – the information conveyed by the nucleus and the satellite in an RST representation is lost.

A similar problem arises when dealing with ‘contrast’ relations. Even though contrasts can be modeled by a parallel structure in a QUD tree, see example (3) Q4.1 and Q4.2, the contrastive meaning...
is not explicitly represented due to the list character of this representation. Thus, it is not impossible to represent a tree that has contrastive relations in a QUD structure in general, but the resulting representation is not explicitly contrastive.

(3) A1: And then there seems to have been some shift in the climate,
{Q2: What did this shift in the climate cause?}
A2': causing Africa to dry in its center,
A2'": that is changing more and more into a savannah landscape.
{Q3: What were the resulting changes in humanity?}
A3: and then there seems to have been a split in the evolution
{Q4: What was the result of this split?}
A4: one part further tried to eat vegetarian food
{Q4.1: What did the first half do?}
A4.1: and the others started to hunt.
{Q4.2: What did the second half do?}
A4.2: and then there seems to have been some shift in the climate,
{Q2: What did this shift in the climate cause?}
A2': causing Africa to dry in its center,
A2'": that is changing more and more into a savannah landscape.
{Q3: What were the resulting changes in humanity?}
A3: and then there seems to have been a split in the evolution
{Q4: What was the result of this split?}
A4: one part further tried to eat vegetarian food
{Q4.1: What did the first half do?}
A4.1: and the others started to hunt.

6.3 Comparison of Speaker Changes

While the QUD model is made to deal with explicit questions and speaker changes in dialog, RST is not. Still, in some examples, both representations deal with the speaker change in a similar way: A different speaker than the person before takes the turn, their utterance has its own subtree giving more information, elaboration, or repeating what has been said, then both models go back to a higher discourse level (see figure 5).

![Figure 5: Representations of the same speaker change in QUD (left) and RST tree (right).](CRE210_Transcript_p2)

In the RST representation, an utterance containing a summary of the previous topic and a transition to the next topic will always be parts of two different sub-trees. On the other hand, in the QUD representation, the transition to the next topic will be a child of the previous topic, as long as both topics are closely connected to each other - see figure 2 for both representations of (4). This means that in the RST representation, the utterance by speaker 1 is split up between two different sub-trees while in the QUD representation, it is not. Unsurprisingly, example (4) has one of the lowest similarities (0.68) in our quantitative comparison between discourse annotations.

Another problem occurs if a speaker change is accompanied by an explicit question. While this is what QUD is meant to model and has no problem dealing with, an explicit question that has no other function than introducing a new topic cannot be appropriately represented in an RST tree without introducing an additional discourse relation.

7 Discussion

In this paper, we have carried out a systematic comparison of the discourse structures induced by RST and QUD frameworks for the same texts from two media, blog posts and podcast transcripts. Our annotations of the 28 texts show that both frameworks
can be successfully applied to monologue texts as well as dialogs.

We compare the branching structure of the resulting RST and QUD trees by first providing a method that converts an RST tree into an equivalent QUD representation. We compare these representations to the manually annotated QUD trees and find an overall similarity of 0.74 – similar to or surpassing the agreement scores between human annotators for discourse structure annotation tasks. This shows that the two frameworks cover some of the same information for our corpus. The similarity between analyses is higher for the blog posts, indicating that the topic structure (QUD) and coherence/intentional structure (RST) are more closely related for monologue texts than for dialog (where two speakers have to agree on how the overall discourse progresses).

Finally, we provide a detailed qualitative comparison of the way complex discourse units are mapped across frameworks, about how certain discourse relations can be represented in QUD trees, and the effects of speaker changes in dialog. We discuss that while the overall structure of QUD and RST trees often matches approximately, QUD trees do not indicate the centrality of discourse segments and cannot represent certain types of relations easily, such as concession and contrast. In contrast, the topic progression within a discourse is captured in QUD analyses but may be missing from RST.

Limitations

We have carried out all analyses according to our best abilities. Nevertheless, it should be noted that RST structures and QUD structures were annotated by distinct researchers. While all annotations have been double-checked by at least one other expert for plausibility, in many cases there are alternative analyses of the texts which may also be applicable (as is usually the case for discourse structure). Since we do not have direct access to the discourse creators and their goals, this limitation is unavoidable in corpus studies.

Ethics Statement

The data reported on in this paper was collected within the research project named below. For all texts, explicit consent was obtained from the creators to use the data for annotation and scientific analysis. The data was automatically and manually preprocessed and reformatted, and manually annotated for discourse structure (among other levels). All annotations were carried out by researchers during their regular work time, either PhD students (authors of this paper) or student research assistants with regular work contracts. The data will be long-term archived and made available to other researchers according to the laws that apply. We see no other ethical issues with our data or research practices.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project ID 317633480 – SFB 1287. We want to thank Elisa Lübbers for helping with RST annotation.

References


Exploiting Knowledge about Discourse Relations for Implicit Discourse Relation Classification

Nobel Jacob Varghese and Frances Yung and Kaveri Anuranjana and Vera Demberg
Language Science and Technology, Saarland University
nobeljacobv@gmail.com, {frances, kaveri, vera}@coli.uni-saarland.de

Abstract

In discourse relation recognition, the classification labels are typically represented as one-hot vectors. However, the categories are in fact not all independent of one another – on the contrary, there are several frameworks that describe the labels’ similarities (by e.g. sorting them into a hierarchy or describing them in terms of features (Sanders et al., 2021)). Recently, several methods for representing the similarities between labels have been proposed (Zhang et al., 2018; Wang et al., 2018; Xiong et al., 2021). We here explore and extend the Label Confusion Model (Guo et al., 2021) for learning a representation for discourse relation labels. We explore alternative ways of informing the model about the similarities between relations, by representing relations in terms of their names (and parent category), their typical markers, or in terms of CCR features that describe the relations. Experimental results show that exploiting label similarity improves classification results.

1 Introduction

Discourse relations (DRs) are logical relations between units of text (“arguments 1 and 2”) that make the whole text coherent, see e.g. the concession relation in (1).

(1) [John prepared for his final exam hoping to get at least a pass.]Arg1 [He got an E.]Arg2 (DR: COMPARISON.CONCESSION_Arg2-ASDENIER)

The task of implicit DR recognition (IDRR) is particularly challenging because informative discourse connectives (DCs), such as “however” are missing. Implicit discourse relation classification tasks using the Penn Discourse Treebank (PDTB) framework (Prasad et al., 2008) typically distinguish between 11 different labels. However, these labels are not completely independent of one another – some relations tend to co-occur or be confused more than others. The similarities between relations are represented in the PDTB relation hierarchy, which groups the labels into four top-level classes, or by the CCR feature representation proposed in Sanders et al. (2021). However, these well-known similarities are typically not exploited for discourse relation classification tasks – instead, all labels are treated as if they were independent of one another.

Guo et al. (2021) recently proposed the Label Confusion Model (LCM), which seems well-suited for the characteristics of the IDRR task: Guo et al. (2021) showed that the method is particularly suitable for problems with many labels, classification problems in which labels are ambiguous and tend to be confused with each other, and/or when there is semantic overlap between the labels. They demonstrated the benefit of the method on several text classification tasks.

The goal of the present paper is to test whether the LCM approach is indeed helpful for IDRR and experiment with three different ways of capturing the label similarities. (1) We use label embeddings: DR labels are not random words but terms that lexically describe the meaning of the DRs, such as REASON, PRECEDENCE, CONDITION, and so on. Using label embeddings assumes that similar relations also tend to have names with similar lexical embeddings. However, some relation labels may additionally be associated with a quite different meaning in normal language use (e.g., “concession”), and their embedding may hence not capture the technical meaning well. (2) We characterize a DR by a set of prototypical connectives (e.g., however and nevertheless for a CONCESSION relation). (3) We encode DRs via their cognitive features (e.g., a concession relation would be described as a negative causal relation).
2 Related work

2.1 Discourse Relation Classification

Our work is not the first to use information from typical connectives for enriching classification: For example, the implicit DCs that are annotated together with the sense labels in PDTB have been incorporated into the training objective (Kishimoto et al., 2020; Jiang et al., 2021a; Kurfalı and Östling, 2021; Jiang et al., 2021b). Several works also utilize the label hierarchy of the PDTB to train the model to learn the difference between the labels by contrastive learning (Long and Webber, 2022) or operate on the label hierarchy for learning sounder embeddings to direct the prediction (Wu et al., 2021). In this work, we operate on the label names to incorporate the information of the DCs and the PDTB hierarchy.

In addition, DRs can be described in terms of features. The Cognitive approach to Coherence Relations (CCR) (Sanders et al., 1992, 2021) characterizes the relation of DRs by “dimensions” such as basic operation, source of coherence, order and polarity. For example, a CONTRAST relation can be described as a negative relation of addition operation with objective source of coherence. We also explore the potential of encoding these unifying dimensions of DRs into the label names for IDRR.

2.2 Exploiting label Similarity

Text classification tasks typically distinguish between a large number of categories or labels. Various approaches have been proposed to model the relation between the semantics of the labels and the text to be classified. Zhang et al. (2018) compares the vectors of the inputs and labels in a multitask learning setting. Wang et al. (2018) use label-based attention scores to embed the label information. Xiong et al. (2021) append the labels to the inputs, such that the embeddings of the labels are learned using the self-attention mechanism of BERT.

Our work builds on the Label Confusion Model (LCM; Guo et al., 2021), which was proposed for learning about the similarity of instances and labels simultaneously during training and which can be expected to be particularly useful in classification tasks with many similar labels. The LCM generates an alternate semantically informed vector in place of one-hot vectors.

For every input to the base model, the LCM inputs all the labels of the corresponding classification tasks, i.e., the LCM is run in parallel with a base model, as seen on the right side of Figure 1. The LCM model consists of a label encoder and a Simulated Label Distribution (SLD) block. The encoder, which comprises an input layer, an embedding layer and a linear layer produces a representation for all the labels.

The representation produced by the base model before the soft-max layer and the representation generated by the LCM encoder is made compatible such that they have dimensions that enable a similarity calculation. A similarity calculation is performed in the SLD block between the representation produced by the base model and the label encoder to generate the SLD distribution in place of one-hot vectors. A controlling parameter is $\alpha$ modulates the balance between the original label one-hot vector and the generated SLD.

Then, KL-divergence loss is computed between the predicted label distribution (PLD) of the base model and the generated SLD. The final labels are predicted using the soft-max classifier of the base model. The LCM trains in parallel with the base model until the LCM-stop epoch, which is determined by a hyper-parameter.

Experiments and analyses on data sets like DB-Pedia1, THUCNews2, etc., show that the LCM can manage to exploit the semantics of the DR labels.

3 Methodology

3.1 LCM for IDRR

We train an 11-way classification model which predicts one of the second-level DR labels defined in PDTB 2.0, as shown in the first column of Table 1, given the two spans of text (called Arg 1 and Arg 2) and the DR links. To do so, we trained the LCM with a state-of-the-art IDRR model, which is the Bilateral Matching and Gated Fusion (BMGF) RoBERTa model (Liu et al., 2020).

The BMGF-RoBERTa is a complex model that comprises six layers: a hybrid representation layer, a context representation layer, a matching layer, a fusion layer, an aggregation layer, and a prediction layer. As shown in Fig 1, the LCM runs concurrently with the BMGF-RoBERTa for each input

1https://www.dbpedia.org/
2http://thuctc.thunlp.org/
instance. We initialize the embedding layer of the LCM with pre-trained GloVe (Pennington et al., 2014) word embeddings of the labels and their variations as described in table 1. The learned representation generated by the prediction layer of the BMGF-RoBERTa is fed as the input to the SLD block of the LCM. KL-divergence loss is calculated between the predicted label distribution (PLD) of the BMGF-RoBERTa and the generated SLD. The final labels are predicted using the soft-max classifier of the BMGF-RoBERTa and the SLD produced by the LCM is utilized for optimizing the loss until the LCM-stop epoch, which is determined by a hyper-parameter. After the LCM-stop epoch, only the BMGF-RoBERTa is trained further and the LCM is inactive.

![Figure 1: Combined architecture of the BMGF-RoBERTa and the LCM](image)

### 3.2 Encoding other DR knowledge

Training with the LCM allows the IDRR model to learn the association between the input arguments and the semantics of the label tokens, such as CONJUNCTION and CAUSE. We hypothesize that more detailed relationships could be learned with more expressive label tokens. We explore three alternative ways of encoding label similarity: via label encodings, via encodings of prototypical connectives and via CCR features.

#### DR labels

The PDTB 2.0 labels are arranged in a three-level hierarchical structure, where the 11-way labels, which are usually used in classification tasks, belong to the second level and are children of one of the four parent categories, namely TEMPORAL, COMPARISON, CONTINGENCY, and EXPANSION. Labels under the same parent category are more closely related than labels of different parent categories. In our experiments, we compare the use of only the level-2 labels with the combination of level-2 and level-1 “parent” labels.

#### Prototypical DCs

DCs are used in both traditional and crowd-sourced annotations to facilitate the identification of the implicit DRs (Prasad et al., 2007; Yung et al., 2019). Most relations can be characterized by some prototypical DCs. For example, a CAUSAL relation is best represented by because and therefore. We define a subset of prototypical DCs for each label and replace the label tokens with the DC tokens. We do not include preposition tokens present in multi-word DCs in order not to dilute the overall semantic representation of the labels (e.g. example instead of for example).

#### Cognitive approach to Coherence Relations (CCR)

Sanders et al. (2021) decompose each third-level DR in the PDTB 2.0 with five unifying dimensions. We specify each second-level relation by the dimension values shared by its children. Two or three dimensions are enough to specify the second-level relations. We use these CRR tokens in addition to the original DR tokens because certain second-level relations, such as CONJUNCTION and RESTATEMENT, have the same set of dimension values. We do not include value tokens that semantically overlap with the relation label. For example, we do not include the value of the temporal order dimension of the SYNCHRONOUS relation, because it is also synchronous.

Table 1 shows the lexical terms we use for each setting. We combine the representation of the multiple tokens per label by summing up the GloVe embeddings of the individual tokens.

#### 3.3 Data and setting

We train and evaluate the proposed model on the PDTB 2.0 data set (Prasad et al., 2008). We use sections 2-20 for the training, 21-22 for testing, and 0-1 for validation, following e.g. Ji and Eisenstein (2015). The models are trained for the 11-way classification of the second-level sense labels.

We use the codes of the BMGF-RoBERTa released on GitHub, which was implemented in PyTorch, and re-implemented the original LCM from the authors' GitHub repository. For integrity, we use single tokens in the original labels. For the PRAGMATIC CAUSE relation, we used the token pragmatic instead of cause since there is already a CAUSE relation. We also experimented with vector averaging. Similar results were obtained.
Table 1: Tokens used in each label representation strategy. The prototype DC tokens replace the original labels while the CCR and parent tokens are used in addition to the original labels.\footnote{The published accuracy of the BMGF-RoBERTa is 58.13. We found that the discrepancy is because, according to the released codes, the result of the best test epoch has been reported. For fair evaluation, we report the results of all the models based on the best validation epoch (based on macro F1).}

<table>
<thead>
<tr>
<th>Original labels</th>
<th>Parent labels</th>
<th>Prototype DCs</th>
<th>CCR features</th>
</tr>
</thead>
<tbody>
<tr>
<td>concession</td>
<td>comparison</td>
<td>despite, even, though, however</td>
<td>negative, causal</td>
</tr>
<tr>
<td>contrast</td>
<td>contingency</td>
<td>contrast, comparison, but</td>
<td>negative, addition, objective</td>
</tr>
<tr>
<td>cause</td>
<td>contingency</td>
<td>because, result, therefore</td>
<td>positive, causal, objective</td>
</tr>
<tr>
<td>pragmatic (cause)</td>
<td>contingency</td>
<td>considering, accordingly</td>
<td>positive, causal, subjective</td>
</tr>
<tr>
<td>alternative</td>
<td>expansion</td>
<td>alternatively, instead, rather</td>
<td>positive, addition</td>
</tr>
<tr>
<td>conjunction</td>
<td>expansion</td>
<td>addition, also, furthermore</td>
<td>positive, addition</td>
</tr>
<tr>
<td>instantiation</td>
<td>expansion</td>
<td>example, instance</td>
<td>positive, addition</td>
</tr>
<tr>
<td>list</td>
<td>expansion</td>
<td>firstly, secondly, thirdly</td>
<td>positive, addition</td>
</tr>
<tr>
<td>restatement</td>
<td>expansion</td>
<td>other, words, means</td>
<td>positive, addition</td>
</tr>
<tr>
<td>asynchronous</td>
<td>temporal</td>
<td>subsequently, afterwards, previously</td>
<td>positive, addition</td>
</tr>
<tr>
<td>synchrony</td>
<td>temporal</td>
<td>same, time, simultaneously, meanwhile</td>
<td>positive, addition</td>
</tr>
</tbody>
</table>

Table 1: Tokens used in each label representation strategy. The prototype DC tokens replace the original labels while the CCR and parent tokens are used in addition to the original labels.\footnote{The published accuracy of the BMGF-RoBERTa is 58.13. We found that the discrepancy is because, according to the released codes, the result of the best test epoch has been reported. For fair evaluation, we report the results of all the models based on the best validation epoch (based on macro F1).}

TensorFlow to PyTorch in order to integrate the two models.

For training, we have utilized 3 × NVIDIA Tesla V100, with a batch size of 16. The pre-trained embedding utilized where GloVe (Pennington et al., 2014) common crawl with 42B tokens. Whenever we utilized the pre-trained word embeddings for the labels, the weights of the embedding layer were frozen and not updated during the training. The values of the hyper-parameters \( \alpha \) is optimized to 4 using initialization in the range of 1–6. The LCM-stop parameter is set to 100, which is chosen based on the implementation of Guo et al. (2021). The results reported below are averaged over five runs.

### 4 Results

Table 2 compares the results of the models evaluated by accuracy and macro F1. It can be observed that all versions of the LCM improved the baseline model. In particular, the LCM model using prototype DCs outperforms the other models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL (Liu et al., 2020)</td>
<td>55.20 (0.013)</td>
<td>36.07 (0.010)</td>
</tr>
<tr>
<td>+ LCM (orig.)</td>
<td>57.20 (0.006)</td>
<td>38.92 (0.014)</td>
</tr>
<tr>
<td>+ LCM (orig.+parent)</td>
<td>57.55 (.010)</td>
<td>40.48 (.006)</td>
</tr>
<tr>
<td>+ LCM (orig.+CCR)</td>
<td>57.69 (.004)</td>
<td>39.45 (.015)</td>
</tr>
<tr>
<td>+ LCM (protyp. DC)</td>
<td>57.80 (0.013)</td>
<td>40.63 (0.025)</td>
</tr>
</tbody>
</table>

Table 2: 11-way classification results on PDTB 2.0.\footnote{The published accuracy of the BMGF-RoBERTa is 58.13. We found that the discrepancy is because, according to the released codes, the result of the best test epoch has been reported. For fair evaluation, we report the results of all the models based on the best validation epoch (based on macro F1).}

Figure 2 compares the distribution of the labels predicted by the baseline and the LCM (protyp. DC) models compared with gold on the five most frequent DR labels. The counts are the average values of the five runs of each model.

![Figure 2: Distribution of the predictions produced by the BL and LCM (protyp. DC) model compared with gold on the five most frequent DR labels. The counts are the average values of the five runs of each model.](image)

The predicted label distributions suggest that the LCM allows the IDR model to learn the difference and similarity between COMPARISON and EXPANSION, but not among different types of EXPANSION. The parent relation and the CCR features of all EXPANSION relations are in fact the same. That could explain why the performances of these two versions on the EXPANSION items are similar, while LCMDC performs slightly better.

\footnote{These are the gold labels of 90% of the test set instances.}
5 Qualitative Analysis

In this section, we analyze some examples that demonstrate that the LCM has better captured the implicit DRs between two arguments.

First, as mentioned in the previous section, the false positive CONTRAST relations predicted by the baseline model are mostly CAUSAL relations. In most of these cases, the Arg2 contains the tokens now or still, which are often used to mark contrast, as in the following example.

(2) [Last week that company and union negotiations had overcome the major hurdle, ...], Arg1 [Now only minor points remain to be cleaned up], Arg2
(gold: : CONTINGENCY, CAUSE
LCM: CONTINGENCY, CAUSE
baseline: COMPARISON, CONTRAST)

In Example 2, the baseline model’s prediction might have been based on the local markers now and the lexical pair major and minor, while the LCM model infers the positive relation between overcome major hurdle and only minor points remain.

Secondly, the LCM models overpredict RESTATEMENT relations, which are annotated as other relations in the PDTB. We found that for some of these cases, a restatement label could actually be justifiable as a secondary label.

(3) [Treating employees with respect is crucial for managers.], Arg1 [It’s in their top five work values.], Arg2
(gold: : CONTINGENCY, CAUSE
LCM: EXPANSION, RESTATEMENT
baseline: COMPARISON, CONTRAST)

(4) [Sotheby’s defends itself and Mr. Paul in the matter.], Arg1 [Mr. Wachter says Mr. Paul was a quick study who worked intensely and bought the best pictures available at the moment.], Arg2
(gold: : EXPANSION, INSTANTIATION
LCM: EXPANSION, RESTATEMENT
baseline: EXPANSION, INSTANTIATION)

In Example (3), respect being crucial is the reason that it is counted as a top value, but these two arguments can also be viewed as different ways to state that it is important for managers to respect their employees. In Example (4), Mr. Wachter’s comment could be an example of how Sotheby’s defends Mr. Paul. However, depending on the context, Arg2 can also be interpreted as a RESTATEMENT. These cases suggest that the LCM tends to confuse relations most easily when they are similar or have semantic overlap.

However, we do note that there are cases where the LCM model indeed overpredicts restatement relations, see example (5).

(5) [It’s no longer enough to beat the guy down the street.], Arg1 [You have to beat everyone around the world.], Arg2
(gold: : EXPANSION, ALTERNATIVE
LCM: EXPANSION, RESTATEMENT
baseline: EXPANSION, ALTERNATIVE)

Finally, comparing the different versions of the LCM models, the LCMDC model outperforms the other two models in predicting CAUSAL and CONJUNCTION relations. A possible explanation is that the DC tokens used to represent these relations are indeed strongly prototypical compared with other relations. This suggests that the choice of prototype DCs has a strong effect on the model performance. On the other hand, the LCMparent model has the highest recall of INSTANTIATION relations, but these are often co-occurring with RESTATEMENT, which is predicted by the other two variants.

6 Conclusion

We proposed to inform an IDR model with knowledge about the DRs encoded in the classification labels using the LCM, instead of treating each class independently. In addition, we explored various strategies to encode different types of knowledge into the model and found that they are all beneficial. This approach is flexible and can also be applied to other base models. Furthermore, learning the lexical semantics of the label tokens allows a model to train on multiple datasets even if they do not share the same label set, and this is the direction of our future work.

7 Limitations

The encoder of the LCM which we have utilized for our experiments is a basic deep neural network. Replacing it with more robust and effective architectures could help achieve better performance. Furthermore, instead of using pre-trained GloVe embeddings for the encoder, using IDDR-specific
embeddings could have been a more efficient approach. Lastly, our models have been trained and evaluated on PDTB 2.0, instead of the latest PDTB 3.0, which includes also intra-sentential implicit relations and has a more systematic sense hierarchy.

Acknowledgments

This project is supported by the German Research Foundation (DFG) under Grant SFB 1102 (“Information Density and Linguistic Encoding”, Project-ID 232722074).

References


Xin Liu, Jiefu Ou, Yangqiu Song, and Xin Jiang. 2020. On the importance of word and sentence representation learning in implicit discourse relation classification.


SAE-NTM: Sentence-Aware Encoder for Neural Topic Modeling

Hao Liu, Jingsheng Gao, Suncheng Xiang, Ting Liu, Yuzhuo Fu*
School of SEIEE, Shanghai Jiao Tong University, China
{liuh236, gaojingsheng, xiangsuncheng17, louisa_liu, yzfu}@sjtu.edu.cn

Abstract

Incorporating external knowledge, such as pre-trained language models (PLMs), into neural topic modeling has achieved great success in recent years. However, employing PLMs for topic modeling generally ignores the maximum sequence length of PLMs and the interaction between external knowledge and bag-of-words (BOW). To this end, we propose a sentence-aware encoder for neural topic modeling, which adopts fine-grained sentence embeddings as external knowledge to entirely utilize the semantic information of input documents. We introduce sentence-aware attention for document representation, where BOW enables the model to attend on topical sentences that convey topic-related cues. Experiments on three benchmark datasets show that our framework outperforms other state-of-the-art neural topic models in topic coherence. Further, we demonstrate that the proposed approach can yield better latent document-topic features through improvement on the document classification.

1 Introduction

Topic models have been widely used to identify human-interpretable topics and learn text representations, which have been applied for various tasks in Natural Language Processing (NLP) such as information retrieval (Lu et al., 2011), summarization (Nguyen et al., 2021), and semantic similarity detection (Peinelt et al., 2020). A typical topic models is based on the latent Dirichlet allocation (LDA) (Blei et al., 2003) and Bayesian inference. However, to avoid the complex and expensive iterative inference of conventional topic models, topic modeling with the deep neural network has been the leading research direction in this field (Miao et al., 2016; Srivastava and Sutton, 2017; Ding et al., 2018).

Neural topic models (NTMs) usually exploit the BOW representation as input, disregarding the syntactic and semantic relationships among the words in a document, thus leading to relatively inferior quality of topics. Recently, pre-trained language models (PLMs) (Kenton and Toutanova, 2019; Reimers and Gurevych, 2019) demonstrate their strong ability to capture sentential coherence by achieving state-of-the-art performance on many natural language processing tasks. Therefore, several approaches have been proposed to incorporate external knowledge into topic models to address the limitations of BOW. A typical method to take external knowledge as additional features (Bianchi et al., 2021; Jin et al., 2021) concentrates the outputs of PLMs with BOW data. Another way (Hoyle et al., 2020) is to distill the knowledge of the teacher PLMs to generate a smoothed pseudo-document, which guides the training of a student topic model.

However, there are still limitations to the above approaches. Firstly, the document-level sequences are too long to be modeled, since the token-level sequence in the context is usually considered as input to the PLMs. Extracting the document-level semantic embedding with PLMs as external knowledge ignores the restriction on sequence length, which loses massive semantic information from input text. Secondly, the difference in learning objectives between NTMs and PLMs makes it challenging to incorporate external knowledge. The encoder of NTMs is designed to handle the sparse BOW data, unable to take into account the dense contextual document embedding from PLMs.

To address these limitations, we build upon the framework of variational autoencoders (VAE) (Kingma and Welling, 2013) and propose a sentence-aware encoder for incorporating external semantic knowledge into topic models. The proposed approach integrates the advantages of NTMs and PLMs as encoders. Specifically, the encoder of the topic model is responsible for processing document-level BOW data like most NTMs, while the PLMs is used to encode sentence-
level semantic information as its original training objective. Different from previous approaches, our proposed framework considers cross-attention (Vaswani et al., 2017) between the BOW data and sentence embeddings, which leverages fine-grained semantic information for topic discovery.

To summarize, the main contributions of this paper are as followed: (1) We propose a novel framework SAE-NTM: Sentences-Aware Encoder for Neural Topic Modeling which leverages the cross-attention for incorporating external semantic knowledge in a sentence-aware manner. (2) Quantitative and qualitative experiments demonstrate that our proposed approach significantly outperforms the existing state-of-the-art topic models in topic coherence. (3) We show that the BOW-guided attention yields practical latent document-topic features, achieving better performance on the document classification task.

2 Methodology

2.1 SAE: Sentence-Aware Encoder

In this section, we introduce the sentence-level semantic information as external knowledge and propose a method to efficiently combine BOW data with external knowledge for document representations, as shown in Figure 1.

Encoder for bag-of-words and sentence sequences. Neural topic models with variational auto-encoders usually take high-dimensional, sparse word counts \( x_i \) as input and transform it into a low-dimensional dense feature \( h_i \), to fit the variational autoencoders framework as formulated in Eq.1.

\[
    h_i = Enc(x_i)
\]

Where \( Enc : \mathbb{R}^V \rightarrow \mathbb{R}^L \) is usually a multi-layer perceptron (MLP) for the inference of the \( i^{th} \) document representation.

Complementary to the orderless BoW, the context of the document carries more affluent and more sophisticated semantic information. And it can be represented as contextual embeddings by pre-trained language models (e.g., BERT (Kenton and Toutanova, 2019)) from large corpora, which have a fine-grained ability to capture aspects of linguistic context. In this paper, we employ sentence-transformers (Reimers and Gurevych, 2019) to encode each sentence in the document as follows:

\[
    \{ h_1^i, \cdots, h_m^i \} = Trans(\{ s_1^i, \cdots, s_m^i \})
\]  

where \( s_j^i \) is a sequence of tokens and \( h_j^i \) is the aggregated contextual embedding from the pre-trained sentence-transformers for the \( j^{th} \) sentence.

Sentence-aware Attention. The contextual embeddings \( \{ h_1^i, \cdots, h_m^i \} \) and BOW representation \( h_i \) jointly constitute the input of sentence-aware encoder. Then sentence-aware attention is employed to accomplish the interaction of word counts and semantic embeddings formulated in Eq.3.

\[
    d_i = \sum_{j=1}^{m} \alpha_j^i h_j^i
\]

\[
    \alpha_j^i = \frac{\exp(score(h_i, h_j^i))}{\sum_{k=1}^{m} \exp(score(h_i, h_k^i))}
\]

Where the representation \( d_i \) of the \( i^{th} \) document is a weighted sum of contextual embeddings \( \{ h_1^i, \cdots, h_m^i \} \) and \( \alpha_j^i \) is the normalized attention of the \( j^{th} \) sentence. Typically, the scoring function \( score \) is scaled dot-product attention. Sentence embeddings as external knowledge provide rich textual information, while the BOW data guides topic model in the assignment of attention on topical sentences, which contributes to capturing the co-occurrence patterns of the words.

2.2 Variational inference

Starting with the document representation, variational inference (Kingma and Welling, 2013) consider Logistic-Normal distribution as the posterior distribution \( q(z \mid x) \), whose mean \( \mu_z \) and variance \( \sigma_z \) vectors are separately derived from the document representation through a linear layer. Then the reparameterization trick in Eq.4 is used to estimate the gradient.

\[
    z_i = \text{softmax}(\mu_i + \sigma_i \cdot \epsilon_i)
\]
Table 1: Results of average NPMI scores with 50 and 200 topics on three datasets. For each group of results, we repeat the experiment five times with different random initialization and report the standard deviation.

<table>
<thead>
<tr>
<th>Method</th>
<th>K=50</th>
<th>K=200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20NG Wiki</td>
<td>IMDb</td>
</tr>
<tr>
<td>W-LDA (Nan et al., 2019)</td>
<td>0.274±0.012</td>
<td>0.134±0.003</td>
</tr>
<tr>
<td>SCHOLAR (Chang et al., 2009)</td>
<td>0.322±0.005</td>
<td>0.167±0.002</td>
</tr>
<tr>
<td>CLNTM (Nguyen and Luu, 2021)</td>
<td>0.327±0.002</td>
<td>0.167±0.002</td>
</tr>
<tr>
<td>CTM (Bianchi et al., 2021)</td>
<td>0.329±0.003</td>
<td>0.170±0.004</td>
</tr>
<tr>
<td>SCHOLAR + BAT (Hoyle et al., 2020)</td>
<td>0.343±0.006</td>
<td>0.301±0.002</td>
</tr>
<tr>
<td>SAE-NTM (Ours)</td>
<td>0.352±0.006</td>
<td>0.314±0.002</td>
</tr>
</tbody>
</table>

where \( \varepsilon_i \sim \mathcal{N}(0, 1) \) denotes samples from the normal distribution and \( z_i \) is the latent document-topic vector. Next, it attempts to reconstruct the original BOW data \( \bar{x}_i \) by modeling the words distributions of topics \( \phi \) as follows:

\[
x_i \sim \text{Multi} \left( \text{softmax} \left( z_i \phi^T \right), \mathcal{N} \right)
\]

where \( \phi \in \mathbb{R}^{V \times K} \) is the word-topic matrix and \( \mathcal{N} \) is a vector of document lengths.

Finally, SAE-NTM are trained by maximizing the Evidence Lower Bound (ELBO) of the marginal likelihood of the BoW data:

\[
L(x) = -E_p \left[ \log p(x | z) \right] + KL(q(z | x) \parallel p(z))
\]

where \( \log p(x | z) \), \( q(z | x) \) and \( p(z) \) are respectively the reconstructed data likelihood, the posterior distribution and prior Dirichlet distribution.

3 Experiments

In this section, we design empirical experiments to answer the following questions of concern in topic modeling. First, how effectively does SAE-NTM perform quantitatively and qualitatively in terms of topic quality? Second, how does SAE-NTM perform in automated document-topic inference for downstream tasks? Besides, more details about the impact of external knowledge on topic modeling can be found in Appendix A.

3.1 Experimental Settings

Datasets. We evaluate our proposed SAE-NTM on three benchmark datasets, which differ significantly in the domain, vocabulary size, and document length: 20Newsgroups (20NG, Lang, 1995)\(^1\), Wikitext-103 (Wiki, Merity et al., 2016)\(^2\), IMDb movie reviews (IMDb, Maas et al., 2011)\(^3\). For consistency with prior work, we adopt the same preprocessing steps and train/dev/test split from the original papers for 20NG (i.e., 48/12/40), Wiki (i.e., 70/15/15), IMDb (i.e., 50/25/25).

Baselines. We compare our model with existing state-of-the-art neural topic models: W-LDA (Nan et al., 2019) is a neural model with wasserstein autoencoder, which approximates the Dirichlet prior by minimizing Maximum Mean Discrepancy. SCHOLAR (Card et al., 2018) is a VAE-based neural topic model with a logistic normal prior by minimizing Maximum Mean Discrepancy. SCHOLAR+BAT (Hoyle et al., 2020) is a knowledge-distilled neural topic model where a BERT-based autoencoder as a teacher provides contextual knowledge for the student model. CTM (Bianchi et al., 2021) is a combined topic model with the incorporation of contextualized document embeddings in neural topic models. CLNTM (Nguyen and Luu, 2021) is a contrastive learning version of the neural topic model through a word-based sampling strategy.

3.2 Evaluation in topic coherence

Since topic models aim to discover a set of latent topics that are meaningful and useful for humans (Chang et al., 2009), we evaluate topic coherence using the Normalized Mutual Pointwise Information (NPMI) which is significantly correlated with human judgments on topic quality (Aletras and Stevenson, 2013; Lau et al., 2014). Specifically, we first select the top 10 words under each topic generated by topic models, and then estimate NPMI scores with reference co-occurrence counts from the held-out corpus, e.g. the dev or test split.

As shown in Table 1, we report the results of the average NPMI over 5 runs with different random seeds for initialization for robustness. It can be observed that our model yields the most coher-
Table 2: Some example topics on three datasets, where the italic words are less relevant to the topic.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>NPMI</th>
<th>Topic Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>20NG</td>
<td>SCHOLAR</td>
<td>0.234</td>
<td>encryption enforcement privacy conversation industry manufacturer protect administration device</td>
</tr>
<tr>
<td></td>
<td>Our Model</td>
<td>0.434</td>
<td>encryption enforcement clipper agency wiretap privacy escrow protect security secure</td>
</tr>
<tr>
<td>Wiki</td>
<td>SCHOLAR</td>
<td>0.379</td>
<td>opera composer repertory theatre conductor libretto operatic painting orchestral painter</td>
</tr>
<tr>
<td></td>
<td>Our Model</td>
<td>0.632</td>
<td>cantata bach recitative oboe continuo soloist chorale viola soprano violin</td>
</tr>
<tr>
<td>IMDb</td>
<td>SCHOLAR</td>
<td>0.161</td>
<td>religious beliefs christian society christ views portray racist issues jesus</td>
</tr>
<tr>
<td></td>
<td>Our Model</td>
<td>0.333</td>
<td>christ christian religion church jesus religious bible faith god beliefs</td>
</tr>
</tbody>
</table>

Table 3: Test Accuracy between different topic models on document classification.

<table>
<thead>
<tr>
<th>Model</th>
<th>20NG</th>
<th>IMDb</th>
</tr>
</thead>
<tbody>
<tr>
<td>W-LDA</td>
<td>52.3</td>
<td>80.3</td>
</tr>
<tr>
<td>SCHOLAR</td>
<td>62.8</td>
<td>82.7</td>
</tr>
<tr>
<td>CLNTM</td>
<td>58.4</td>
<td>79.5</td>
</tr>
<tr>
<td>CTM</td>
<td>62.4</td>
<td>84.5</td>
</tr>
<tr>
<td>SCHOLAR + BAT</td>
<td>65.2</td>
<td>83.1</td>
</tr>
<tr>
<td>SAE-NTM (Ours)</td>
<td>66.1</td>
<td>85.9</td>
</tr>
</tbody>
</table>

We report classification accuracy on the test split of 20NG and IMDb in Table 3. It is worth noting that we aim to evaluate the predictive capability of topic models by the performance in document classification, rather than training the model to obtain higher accuracy. The document-topic features provided by our proposed model achieve best accuracy for all the datasets with a significant improvement. It demonstrates that the proposed sentence-aware encoder not only discovers topics that are more meaningful to humans, but also learns better latent document features.

4 Conclusions

In this paper, we propose a Sentence-Aware Encoder for Neural Topic Modeling framework: SAE-NTM to incorporate external knowledge into neural topic models. The proposed method can capture document information by performing attention on sequential sentences in a bag-of-words guided manner. Extensive experiments have shown that our framework can achieve state-of-the-art performance in topic coherence and encode better latent document-topic features. In the future, we would like to explore the possibility of integrating our approach with neural topic models built on other frameworks, such as generative adversarial training (Nan et al., 2019; Wang et al., 2020).
Limitations

The proposed model with sentence-aware encoder aims to efficiently incorporate external knowledge and bag-of-words for topic modeling, which means that in this work we are mainly interested in how documents should be encoded for topic inference. However, the decoder of topic models can also be coupled with word embeddings through factorization, such as embedded topic models (Dieng et al., 2020). It is worth exploring how hierarchical semantic embeddings can be employed for topic modeling with our model.

In this paper, we do not conduct any fine-tuning for the pre-trained language model. Our approach reveals how the frozen pre-trained language model can be effectively used to improve the performance of the topic model with limited computational overhead, given that the parameter size of the pre-trained language model is much larger than that of the topic model. Moreover, fine-tuning pre-trained language models for topic modeling as an unsupervised learning task (Mueller and Dredze, 2021) is challenging.

References


Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.


A Analysis on individual topics

To evaluate whether our improvements are meaningful on individual topics, we directly compare each of the aligned topics generated by the baseline SCHOLAR without external knowledge and our model. Follow previous works (Hoyle et al., 2020; Nguyen and Luu, 2021), we align the topics by using a variation of competitive linking to greedily approximate the optimal weight of the bipartite graph matching. And the weight of each link is calculated based on the similarity between their word distributions as measured Jenson-Shannon (JS) divergence (Wong and You, 1985; Lin, 1991). We iteratively select the topic pair with the lowest score based on JS divergence, separate the two topics from the topic list, and repeat until the rest JS score exceeds a certain threshold.

Figure 2 shows the JS-divergences for aligned topic pairs for three benchmark corpora. Based on visual inspection, we choose the most aligned 44 topic pairs to conduct the comparison, since there is no conceptual relationship between topic pairs beyond this point and employ the same threshold across all three datasets for simplicity. Considering these aligned topic pairs conceptually related, we explore the impact of external knowledge on the baseline topic model on a topic-by-topic basis as shown in Figure 3. It can be observed that the number of topics with high NPMI scores from our model is apparently more than that of the baseline model. This means that the overall promotion achieved by our approach can be interpreted as identifying the topic space generated by the baseline models and in most cases, improving the coherence of individual topics.

Figure 2: Jensen-Shannon divergence for aligned topic pairs of SCHOLAR and our model.

Figure 3: The number of aligned topic pairs which our model improves upon SCHOLAR model.
Abstract

Document-level context for neural machine translation (NMT) is crucial to improve the translation consistency and cohesion, the translation of ambiguous inputs, as well as several other linguistic phenomena. Many works have been published on the topic of document-level NMT, but most restrict the system to only local context, typically including just the one or two preceding sentences as additional information. This might be enough to resolve some ambiguous inputs, but it is probably not sufficient to capture some document-level information like the topic or style of a conversation. When increasing the context size beyond just the local context, there are two challenges: (i) the memory usage increases exponentially (ii) the translation performance starts to degrade. We argue that the widely-used attention mechanism is responsible for both issues. Therefore, we propose a constrained attention variant that focuses the attention on the most relevant parts of the sequence, while simultaneously reducing the memory consumption. For evaluation, we utilize targeted test sets in combination with novel evaluation techniques to analyze the translations in regards to specific discourse-related phenomena. We find that our approach is a good compromise between sentence-level NMT vs attending to the full context, especially in low resource scenarios.

1 Introduction

Machine translation (MT) is the task of mapping some input text onto the corresponding translation in the target language. MT systems typically operate on the sentence-level and utilize neural networks trained on large amounts of bilingual data (Bahdanau et al., 2014; Vaswani et al., 2017). These neural machine translation (NMT) systems perform remarkably well on many domains and language pairs, sometimes even on par with professional human translators. However, when the automatic translations are evaluated on the document-level (e.g. the translation of a whole paragraph or conversation is evaluated), they reveal shortcomings regarding consistency in style, entity-translation or correct inference of the gender, among other things (Läubli et al., 2018; Müller et al., 2018; Thai et al., 2022). The goal of document-level NMT is to resolve these shortcomings by including context information as additional input when translating a sentence.

In recent years, many works have been published on the topic of document-level NMT. However, most of these works focus only on including a few surrounding sentences as context. When the context size is increased beyond that, typically a degradation of overall translation performance is reported. Additionally, the transformer architecture as the quasi standard in NMT seems sub optimal to handle long sequences as input/output, since the memory complexity increases quadratically with the sequence length. This is due to the attention mechanism, where each token in a sequence needs to attend to all other tokens.

In this work, we propose a constrained attention variant for the task of document-level NMT. The idea is to reduce the memory consumption while at the same time focusing the attention of the system onto the most relevant parts of the sequence. Our contributions are two-fold:

1. We observe that the attention patterns become less focused on the current sentence when increasing the context-size of our document-level NMT systems. Therefore we propose a constrained attention variant that is also more memory efficient.

2. We utilize a targeted evaluation method to assess automatic translations in regards to consistency in style and coreference resolution. We find that our document-level NMT approach performs among the best across all language-pairs and test scenarios.
2 Related Work

Many works have been published on the topic of document-level NMT. The widely used baseline approach consists of simply concatenating a few adjacent sentences and feeding this as an input to the MT system, without modifying the system architecture in any way (Tiedemann and Scherrer, 2017; Bawden et al., 2018; Agrawal et al., 2018; Talman et al., 2019; Nguyen et al., 2021; Majumder et al., 2022). Also, several modifications to this baseline concatenation approach have been proposed. Ma et al. (2020) introduce segment embeddings and also partially constrain the attention to the tokens of the current sentence. Zhang et al. (2020) propose to calculate the self-attention both on the sentence- and on the document-level and then combine the two representations. Fernandes et al. (2021) and Lei et al. (2022) both mask out tokens in the current sentence to increase context utilization while Yang et al. (2023) remove tokens from the context if they are not attended. Typically, slight improvements in BLEU are reported as well as more significant improvements on targeted test sets e.g. for coreference resolution.

Apart from the simple concatenation method, there exist other approaches to document-level NMT. They include using a single document-embedding vector (Macé and Servan, 2019; Stojanovski and Fraser, 2019; Jiang et al., 2020; Huo et al., 2020), multiple encoders (Jean et al., 2017; Bawden et al., 2018; Wang et al., 2017; Voita et al., 2018; Zhang et al., 2018), hierarchical attention (Miculicich et al., 2018; Maruf et al., 2019; Tan et al., 2019; Wong et al., 2020), translation caches (Maruf and Haffari, 2018; Tu et al., 2018; Kuang et al., 2018) or dynamic evaluation (Masimov et al., 2021). However, these approaches are less versatile and require significant changes to the model architecture, often introducing a significant amount of additional parameters. Furthermore, recent works have concluded that the baseline concatenation approach first proposed by Tiedemann and Scherrer (2017) performs as good - if not better - than these more complicated approaches (Lopes et al., 2020; Sun et al., 2022).

While the concatenation approach works well for short context sizes, when used with a larger number of context sentences, typically performance degradation is reported: Scherrer et al. (2019) saw a severe performance degradation when using input sequences with a length of 250 tokens. Liu et al. (2020) could not get their system to converge when using context sizes of up to 512 tokens. They improve training stability by adding additional monolingual data via pre-training. Bao et al. (2021) also report that their systems with context length of more than 256 tokens fail to converge. They propose to partially constrain the attention to the current sentence, similar to Zhang et al. (2020). Sun et al. (2022) try to translate full documents with the concatenation approach but could not get their system to converge during training. Their solution is to mix document- and sentence-level data, which reportedly improves system convergence. Li et al. (2022) report severe performance degradation for context sizes longer than 512 tokens. They argue this is due to insufficient positional information and improve performance by repeatedly injecting this information during the encoding process. However, increasing the context size seems to not always result in performance degradation. In their works, Junczys-Dowmunt (2019) and Saleh et al. (2019) train systems with a context size of up to 1000 tokens without degradation in translation quality, which stands in contrast to the works mentioned above and which we will discuss again in the context of our own results. We want to point out that all of the approaches mentioned above still have the problem of quadratically increasing resource requirements, which poses a big challenge even on modern hardware.

Since our proposed approach consists of modifying the attention matrix in the model architecture, we give a brief overview of previous works related to this concept. The works of Ma et al. (2020), Zhang et al. (2020) and Bao et al. (2021) are most closely related and were already mentioned above. All three papers restrict the attention (partially) to the current sentence and combine sentence- and document-level attention context vectors for the final output. However, this means all approaches still suffer from the quadratic dependency on the number of input tokens. Luong et al. (2015) were among the first to propose using the attention concept for the task of MT. They also proposed using a sliding-window with target-to-source alignment for attention similar to us. However, they only work on sentence-level NMT and to the best of our knowledge, this approach was never before transferred to document-level NMT. Shu and Nakayama (2017) and Chen et al. (2018) both extend the approach of Luong et al. (2015) while still working...
solely on sentence-level NMT. Our approach is also related to the utilization of relative positional encoding, which was introduced by Shaw et al. (2018) and later extended by Yang et al. (2018) to be applicable for cross-attention. The work by Indurthi et al. (2019) should also be mentioned, where they pre-select a subset of source tokens on which to perform attention on. Again, all of the above mentioned works only perform experiments on sentence-level NMT. The works of Child et al. (2019), Sukhbaatar et al. (2019) and Guo et al. (2020) are also related, since they use attention windows similar to us for tasks other than MT.

Finally, we briefly want to touch on the subject of automatic evaluation of document-level MT systems. Many works only report results on general MT metrics like BLEU (Papineni et al., 2002), sometimes matching n-grams across sentence-boundaries. However, it has been argued that these metrics do not capture well the very specific improvements that could be expected by including document-level context and that the reported improvements rather come from regularization effects and comparing to sub optimal baseline performance (Kim et al., 2019; Li et al., 2020; Nguyen et al., 2021). Several targeted test suites have been released to better assess the improvements gained by document-level NMT (Müller et al., 2018; Baden et al., 2018; Voita et al., 2019; Jwalapuram et al., 2019). These test suites have some limitations, for example they are language-specific and they are based on just scoring predefined contrastive examples without scoring the actual translations. More recently, Jiang et al. (2022) and Currey et al. (2022) have released frameworks that allow to score the actual MT hypotheses in regards to their consistency regarding specific aspects of the translation.

3 Methodology

Here, we explain the baseline concatenation approach (Section 3.1), the more refined method that we are comparing ourselves against (Section 3.2) as well as our own approach (Section 3.3). We also discuss our different evaluation approaches in Section 3.5.

3.1 The Baseline Concatenation Approach

The baseline concatenation approach is very simple and follows Tiedemann and Scherrer (2017) using the vanilla transformer architecture (Vaswani et al., 2017). Assume we are given a document $D = (F_n, E_n)$ consisting of $N$ source-target sentence-pairs $(F_n, E_n)$. If we want our model to have a context length of $k$ sentences, we simply concatenate the current input sentence with its $k-1$ predecessor sentences and the input to the model would be

$$F_{n-k} <\text{sep}> F_{n-k+1} \ldots <\text{sep}> F_n <\text{eos}>$$

while on the target side we include the preceding sentence as a prefix

$$E_{n-k} <\text{sep}> E_{n-k+1} \ldots <\text{sep}> E_{n-1} <\text{sep}>.$$  

We use a special token $<\text{sep}>$ as a separator between adjacent sentences and $<\text{eos}>$ denotes the end of the sequence. This is done to make it easier for the model to distinguish between the sentence that needs to be translated and the context. Furthermore, we use a special token $F_0 = E_0 = <\text{bod}>$ to denote the start of a document. Since we use the vanilla transformer architecture with self-attention and cross-attention components, the memory usage is $O(L^2)$ with $L$ being the sequence length.

When we train full document-level systems, we simply concatenate all sentences in the document using again the special $<\text{sep}>$ token. Due to hardware limitations, if the length of the target-side of the document exceeds 1000 tokens, we split the document into smaller parts of roughly equal length (i.e. a document of length 1500 tokens would be split into two parts with ca. 750 tokens each).

In a preliminary study, we train systems using no context (sentence-level), just a single sentence as context as well as the maximum context size of 1000 tokens. When looking at the percentage of attention that is payed to the $n$-th source sentence $F_n$ when decoding the $n$-th target sentence $E_n$ (extracted from cross-attention module, see Table 1) we find that this percentage becomes lower as the context size increases. This finding motivates us to explore approaches that bias the attention towards the current sentence.

<table>
<thead>
<tr>
<th>Model</th>
<th>Context</th>
<th>Attn. [%]</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>sent.-level</td>
<td>0 sent.</td>
<td>100.0</td>
<td>32.8</td>
</tr>
<tr>
<td>concat adj.</td>
<td>1 sent.</td>
<td>76.0</td>
<td>33.1</td>
</tr>
<tr>
<td></td>
<td>1000 tok.</td>
<td>46.6</td>
<td>29.5</td>
</tr>
</tbody>
</table>

Table 1: Percentage of attention on the $n$-th source sentence during decoding the $n$-th target sentence, as well as overall translation quality measured in BLEU, for the newstest2018 test set of the NEWS task.
3.2 LST-attention

This method was proposed by Zhang et al. (2020) and is called Long-Short Term (LST) attention. The authors find that their approach outperforms the baseline concatenation approach but they only use a maximum of 3 sentences as context. Nevertheless we deem this approach promising, since it also focuses the attention onto the current sentence. The input to the system is augmented in the same way as described in Section 3.1. Given some queries \( Q \in \mathbb{R}^{I \times d} \), keys \( K \in \mathbb{R}^{J \times d} \) and values \( V \in \mathbb{R}^{J \times d} \), Zhang et al. (2020) formulate their restricted version of the attention as

\[
A(Q, K, V) = \text{softmax} \left( \frac{Q \cdot K^\top}{\sqrt{d}} + M \right) V
\]

with \( d \) being the hidden dimension of the model and \( M \in \mathbb{R}^{I \times J} \) being the masking matrix. This masking matrix is defined as

\[
M_{i,j} = \begin{cases} 
0 & , s(i) = s(j) \\
-\infty & , \text{otherwise}
\end{cases}
\]

where \( s(\cdot) \in \{1, \ldots, N\} \) is a function that returns the sentence index that a certain position belongs to. This means we are restricting the attention to be calculated only within the current sentence. For self-attention in the encoder and the decoder, Zhang et al. (2020) calculate both the restricted and the non-restricted variant and then combine the output context-vectors via concatenation and a linear transformation. The cross-attention between encoder and decoder remains unchanged in this approach and the memory consumption remains \( \mathcal{O}(L^2) \).

3.3 window-attention

This method is proposed by us. We can use the same formulation as above to describe this approach by simply changing the definition of the attention mask to

\[
M_{i,j} = \begin{cases} 
0 & , b_i - w \leq j \leq b_i + w \\
-\infty & , \text{otherwise}
\end{cases}
\]

where \( w \) is the window size and \( b_i \in \{1, \ldots, J\} \) is a target-source alignment. This means a certain query vector \( q_i \) is only allowed to attend to the key vectors \( k_j \) that surround the position \( b_i \) that this query vector is aligned to. We replace all self-attention and cross-attention modules in our network with this window-attention variant. Please note that in practice we do not calculate this mask, but instead we first select the corresponding key-vectors for each query and then calculate attention only between these subsets which reduces the memory consumption from \( \mathcal{O}(L^2) \) to \( \mathcal{O}(L \cdot w) \). We also want to point out that with this approach, the context is not as restricted as it seems on first glance. For any individual attention module, the context is restricted to \( 2 \cdot w \) or \( w \) for self-attention in the encoder and decoder respectively. However, since in the transformer architecture we stack multiple layers, we get a final effective context size of \( 2 \cdot w \cdot \text{num_enc_layers} + w \cdot \text{num_dec_layers} \).

This approach requires us to define an alignment function \( b_i : [1, J] \rightarrow [1, J] \). For self-attention, we assume a 1-1 alignment so the alignment function is the identity function \( b_i = i \). For cross-attention, during training we use a linear alignment function

\[
b_i = \text{round}(J \cdot i)
\]

where \( J \) is the number of tokens in the source document and \( I \) is the number of tokens in the target document. This is not possible during decoding, as we do not know the target document length beforehand. Therefore, we propose three different ways to approximate the alignment during decoding:

1. 1-1 alignment: \( b_i = i \)

2. linear alignment: \( b_i = \text{round}(\text{train_ratio} \cdot i) \)

where we define \( \text{train_ratio} \) as the average source-target ratio over all documents in the training data.

3. sent-align: assume we have already produced \( N' \) full target sentences (i.e. we have produced \( N' \) \(<\text{sep}>\) tokens) up to this point, then

\[
b_i = \begin{cases} 
\sum_{n=1}^{N'} J_n + 1 & , e_{i-1} == <\text{sep}> \\
b_{i-1} + 1 & , \text{otherwise}
\end{cases}
\]

with \( J_n \) being the length of the \( n \)-th source sentence in the input document. In simple terms, when starting to decode a new sentence, we always force-align to the beginning of the corresponding source sentence.

We also test the window-attention approach with relative positional encoding in the self-attention.
instead of absolute positional encoding, which in this framework only requires a small modification to Equation 1:

\[ M_{i,j} = \begin{cases} r_{i-j} & b_i - w \leq j \leq b_i + w \\ -\infty & \text{otherwise} \end{cases} \]

where \( r_{i-j} \in \mathbb{R}^d \) are additional learnable parameters of the network.

### 3.4 Decoding

During decoding, given a document \( F_1^N \), we want to find the best translation \( \hat{E}_1^N \) according to our model. We can not perform exact search due to computational limitations, therefore we have to use approximations. There exist multiple approaches for decoding with a document-level NMT system and since we could not determine a single best approach from the literature, we describe and compare two competing approaches.

**Full Segment Decoding (FSD)** (Liu et al., 2020; Bao et al., 2021; Sun et al., 2022): we split the document into non-overlapping parts \( F_{1}^{k}, F_{k+1}^{2k}, \ldots, F_{N-k}^{N} \) and translate each part separately using

\[ \hat{E}_{i-k} = \arg\max_{E_{i-k}} \{ p(E_{i-k}^i | F_{i-k}^i) \} , \]

which is approximated using standard beam search on the token level (we use beam size 12 for all experiments). For the full document-level systems, we simply use

\[ \hat{E}_1^N = \arg\max_{E_1^N} \{ p(E_1^N | F_1^N) \} . \]

**Sequential Decoding (SD)** (Miculicich et al., 2018; Voita et al., 2019; Garcia et al., 2019; Fernandes et al., 2021): we generate the translation sentence by sentence, using the previously generated target sentences as context:

\[ \hat{E}_i = \arg\max_{E_i} \{ p(E_i | \hat{E}_{i-k}^{i-1}, F_{i-k}^i) \} . \]

### 3.5 Evaluation

For all tasks we report BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) using the SacreBLEU (Post, 2018) toolkit. In addition, for the two En→De tasks (NEWS and OS) we analyze the translations in regards to ambiguous pronouns and style. For pronouns, the goal is to measure how well a system can translate the English 3rd person pronoun ‘it’ (and its other forms) into the correctly gendered German form (which can be male, female or neuter depending on the context). For style, the goal is to measure, how well a system can translate the 2nd person pronoun ‘you’ (and its other forms) into the correct style in German. For example, ‘you’ (singular) can be translated into ‘sie’ or ‘du’ in German, depending if the setting is formal or informal. We employ several strategies to determine the systems ability to disambiguate these phenomena.

We utilize the ContraPro test suite (Müller et al., 2018) and report the contrastive scoring accuracy for pronoun resolution. The test suite contains 12,000 English sentences, each with the correct German reference as well as 2 contrastive German references where the pronoun has been changed to a wrong gender. We score all test cases with the NMT system and each time the system gives the best score to the true reference, it gets a point. In the end we report the scoring accuracy, i.e. the number of points the system has gathered divided by 12,000.

Additionally we also report F1 scores for pronoun and style translation, the method for which is inspired by Jiang et al. (2022). We use parts of speech (POS) taggers as well as language-specific regular expressions to identify ambiguous pronouns/formality in the test sets. We then compare the occurrences in the reference against the occurrences in the hypothesis, calculate precision and recall and then finally the F1 score for both pronoun as well as formality translation. The exact algorithm as well as detailed data statistics for the test sets are given in Appendix A.1.

We report pronoun translation F1 score for both NEWS and OS tasks and the formality translation F1 score only for the OS task, since in the NEWS test set there are not enough examples of ambiguous formality cases. Our extension to the work of Jiang et al. (2022) can be found here: [https://github.com/christian3141/BlonDe](https://github.com/christian3141/BlonDe).

### 4 Experiments

We perform experiments on three document-level translation benchmarks. We call them NEWS (En→De) with newstest2018 as test set, TED (En→It) with tst2017 as test set and OS (En→De) where the test set is simply called test. NEWS is a collection of news articles, TED is a collection of
transcribed TED talks and their respective translations and OS consists of subtitles for movies and TV shows. Especially the latter holds many examples for discourse between different entities. For the details regarding data conditions, preparation and training, we refer to Appendix A.2.

4.1 GPU Memory efficiency

First, we compare the GPU memory consumption of the baseline concat-adj. approach against the window-attention approach for various input sequence lengths. The results are shown in Table 2. As expected, the memory usage increases at a much higher rate for the concat-adj. approach, while the window-attention approach scales roughly linearly, the slope being a function of the window-size $w$.

<table>
<thead>
<tr>
<th># target tokens</th>
<th>concat-adj. $w = 10$</th>
<th>window-attn $w = 10$</th>
<th>window-attn $w = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>736</td>
<td>2.3 GB</td>
<td>2.4 GB</td>
<td>3.5 GB</td>
</tr>
<tr>
<td>1472</td>
<td>5.8 GB</td>
<td>3.9 GB</td>
<td>5.9 GB</td>
</tr>
<tr>
<td>2208</td>
<td>10.9 GB</td>
<td>5.2 GB</td>
<td>8.5 GB</td>
</tr>
</tbody>
</table>

Table 2: GPU-memory consumption for the different approaches when training on a single document of specified number of target tokens.

In terms of positional encoding, relative works significantly better than absolute for the window-attention system. We also test relative positional encoding (window-size 20) for the baseline concat-adj. method, but here the training did not converge. This is, because for long input sequences the system without explicit target-source alignment can no longer distinguish the token ordering on the source side (on the target side it is still possible due to the causal attention mask). The only way to resolve this would be to drastically increase the window-size for the relative positions, however, this would add a significant amount of additional parameters to the network so we decide against this. In terms of alignment, using the sent-align variant significantly outperforms the other approaches. For the window-size, 20 works best. An important finding is, that if we make the window too large, we start losing performance, probably due to the less focused attention problem discussed in Section 3.1.

4.2 Comparison of Decoding Strategies

After training all models on the NEWS task according to Appendix A.2, we test the different search strategies for each of the systems, the result of which can be found in Table 3. For the baseline concat-adj. approach as well as the LST-attn approach, FSD works best. However, we still see significant performance degradation for the systems using long context information. For concat-adj. and LST-attn with 1000 tokens context size, SD performs very poorly. This is because when beginning translating a document, the input sequences are very short and the systems can not appropriately handle that. However, FSD sometimes leads to sentence-misalignment while translating a document, resulting in a lower BLEU score as well. For the window-attention approach (rel. pos. enc., sent-align, window-size 20) we find that the SD decoding strategy works best. Since this approach seems to be able to better handle short input sequences, SD performs better than FSD, since it seems more robust to sentence-misalignment. Moving forward, all numbers reported will be generated with the best respective decoding approach, i.e. SD for window-attention and FSD for all other approaches.

4.3 Hyperparameter Tuning

Our window-attention approach has three hyperparameters that need to be tuned: (i) positional encoding variant (ii) alignment variant during search (iii) window size. Again, we use the NEWS task for tuning and the results for the different variants can be found in Table 4.

In terms of positional encoding, relative works significantly better than absolute for the window-attention system. We also test relative positional encoding (window-size 20) for the baseline concat-adj. method, but here the training did not converge. This is, because for long input sequences the system without explicit target-source alignment can no longer distinguish the token ordering on the source side (on the target side it is still possible due to the causal attention mask). The only way to resolve this would be to drastically increase the window-size for the relative positions, however, this would add a significant amount of additional parameters to the network so we decide against this. In terms of alignment, using the sent-align variant significantly outperforms the other approaches. For the window-size, 20 works best. An important finding is, that if we make the window too large, we start losing performance, probably due to the less focused attention problem discussed in Section 3.1.

4.4 Final Performance Comparison

In Table 5 we report the translation performance of the different document-level approaches on all three translation benchmarks measured in terms of BLEU and TER. None of the document-level systems can consistently outperform the sentence-
Table 4: Results for the different hyperparameter settings of the window-attention system reported on the newstest2018 test set of the NEWS task. All systems have context size 1000 tokens.

<table>
<thead>
<tr>
<th>Model</th>
<th>pos. enc.</th>
<th>Alignment</th>
<th>window-size</th>
<th>BLEU</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>concat-adj.</td>
<td>abs.</td>
<td>-</td>
<td>-</td>
<td>29.5</td>
<td>53.7</td>
</tr>
<tr>
<td></td>
<td>rel.</td>
<td>-</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>window-attn</td>
<td>abs. 1-1</td>
<td>20</td>
<td>29.7 51.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>train avg.</td>
<td>20</td>
<td>28.1 55.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sent-align</td>
<td>10</td>
<td>28.3 53.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>30.3 50.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>29.4 52.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rel. 1-1</td>
<td>20</td>
<td>31.9 49.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>train avg.</td>
<td>20</td>
<td>30.5 53.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sent-align</td>
<td>10</td>
<td>30.6 51.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>33.1 48.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>32.8 48.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Results for the different document-level approaches in terms of BLEU and TER on the three translation benchmarks. Best results for each column are highlighted. External baselines are from † Kim et al. (2019), ‡ Yang et al. (2022) and * Huo et al. (2020).

<table>
<thead>
<tr>
<th>Model</th>
<th>Context</th>
<th>NEWS</th>
<th>TED</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>newstest2018</td>
<td>tst2017 test</td>
<td>test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLEU</td>
<td>TER</td>
<td>BLEU</td>
</tr>
<tr>
<td>sent.-level (external)</td>
<td>0 sent.</td>
<td>32.3</td>
<td>-</td>
<td>33.4</td>
</tr>
<tr>
<td></td>
<td>32.8 49.0</td>
<td></td>
<td></td>
<td>34.2 46.3</td>
</tr>
<tr>
<td>sent.-level (ours)</td>
<td>2 sent.</td>
<td>33.4 48.6</td>
<td>34.3 46.3</td>
<td>38.2 43.9</td>
</tr>
<tr>
<td></td>
<td>1000 tok.</td>
<td>29.5 53.7</td>
<td>32.1 48.4</td>
<td>38.1 46.0</td>
</tr>
<tr>
<td>concat adj.</td>
<td>1000 tok.</td>
<td>30.0 53.1</td>
<td>29.8 54.5</td>
<td>38.5 45.1</td>
</tr>
<tr>
<td>LST-attn</td>
<td>1000 tok.</td>
<td>33.1 48.1</td>
<td>34.6 45.8</td>
<td>38.3 44.4</td>
</tr>
</tbody>
</table>

level baseline on all tasks. On the OS test set, there is a disagreement between BLEU and TER which we think comes from the fact that the average sentence-length on this test set is quite short. The hypothesis of the sentence-level system is the shortest of all hypotheses and also shorter than the reference which gets punished more heavily by BLEU than TER. Out of all full-document approaches, window-attention performs best and is on par with the sentence-level baseline and the document-level system using only 2 sentences as context. For full-document translation, LST-attn performs better than the baseline concatenation approach but still falls behind the sentence-level system especially on the NEWS and TED tasks. One possible reason for why these approaches work better on OS is, that for this task we have much more training data available than for NEWS and TED. We argue that this could also be the reason for the conflicting results reported by Junczys-Dowmunt (2019) and Saleh et al. (2019) compared to the other works who report performance degradation for longer context sizes (see Section 2). However, we leave a detailed analysis of this for future work.

Next, we analyze the ability of the systems to translate ambiguous pronouns and to translate in a consistent style using the methods explained in Section 3.5. The results for the two En→De tasks can be found in Table 6. For both NEWS and OS, all document-level systems can significantly improve over the sentence-level baseline in terms of pronoun translation. We also find that a context longer than two sentences does not seem to help for the pronoun task. This is actually to be expected since typically the distance between noun and pronoun is not that large and according to Müller et al. (2018), the overwhelming majority of ContraPro test cases do not require more than two sentences as context. For the correct translation of the style however, the larger context size is clearly beneficial.
Table 6: Results for the different document-level approaches in terms of pronoun and formality translation. Best results for each column are highlighted.

<table>
<thead>
<tr>
<th>Model</th>
<th>Context</th>
<th>NEWS</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BALEU</td>
<td>Scoring</td>
</tr>
<tr>
<td>sent.-level</td>
<td>0 sent.</td>
<td>18.4</td>
<td>48.2</td>
</tr>
<tr>
<td>concat adj.</td>
<td>2 sent.</td>
<td>19.6</td>
<td>67.9</td>
</tr>
<tr>
<td>window-attn</td>
<td>1000 tok.</td>
<td>16.8</td>
<td>61.9</td>
</tr>
<tr>
<td>LST-attn</td>
<td>1000 tok.</td>
<td>19.6</td>
<td>63.0</td>
</tr>
</tbody>
</table>

Table 7: Example translation of a snippet from the OpenSubtitles test set. Formal 2nd person pronouns are marked in red and informal ones are marked in blue.

<table>
<thead>
<tr>
<th>source reference</th>
<th>target reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Was zwischen Ihnen und Dr. Webber ist, geht mich nichts an...</td>
<td>Was zwischen dir und Dr. Webber ist, geht mich nichts an...</td>
</tr>
<tr>
<td>- Du schuldest mir keine Entschuldigung.</td>
<td>- Du schuldest mir keine Entschuldigung.</td>
</tr>
<tr>
<td>Sie schulden Dr. Bailey etwas.</td>
<td>- Du schuldest Dr. Bailey eine.</td>
</tr>
<tr>
<td>Wir haben für Dr. Webber Partei ergriffen.</td>
<td>Wir haben uns für Dr. Webber eingesetzt.</td>
</tr>
<tr>
<td>Ich verstehe nicht warum...</td>
<td>Ich verstehe nicht warum...</td>
</tr>
<tr>
<td>Dr. Webber braucht Sie nicht, um seine Schlachten zu schlagen.</td>
<td>Dr. Webber braucht dich nicht, um seine Schlachten zu kämpfen.</td>
</tr>
<tr>
<td>Was du getan hast, verletzt das gesamte Krankenhaus.</td>
<td>Was du getan hast, hat dem ganzen Krankenhaus geschadet.</td>
</tr>
<tr>
<td>Ihre oberste Priorität muss dieser Ort und seine Patienten sein.</td>
<td>Deine oberste Priorität muss diesem Haus und seinen Patienten gelten.</td>
</tr>
</tbody>
</table>

5 Conclusion

In this work, we focus on methods to increase the context-size for document-level NMT systems. We point out the shortcomings of the baseline approaches to long-context document-level NMT and in turn propose to modify the attention component to be more focused and also to be more memory efficient. We compare our approach against approaches from literature on multiple translation tasks and using different targeted evaluation methods. We confirm the improved memory efficiency of the proposed method. We find that for some discourse phenomena like pronoun translation, the longer context information is not necessary. For other aspects, like consistent style translation, the longer context is very beneficial. It seems that the baseline concatenation approach needs large amounts of training data to perform well for larger context sizes. We conclude that our approach performs among the best across all tasks and evaluation methods, with the additional benefit of reduced memory consumption for long input sequences.
Acknowledgements

This work was partially supported by the project HYKIST funded by the German Federal Ministry of Health on the basis of a decision of the German Federal Parliament (Bundestag) under funding ID ZMV11-2520DAT04A, and by NeuroSys which, as part of the initiative “Clusters4Future”, is funded by the Federal Ministry of Education and Research BMBF (03ZU1106DA).

Limitations

This work is about document-level NMT, we focus specifically on methods that improve the model performance for long input sequences. Due to constrained resources, this work has several limitations. To be able to train all methods including the inefficient baseline approach, we have to limit the context size to 1000 tokens. While we do a comparison to existing approaches, other approaches have been proposed to improve the performance of systems with long context information, which we do not compare against. We run experiments on three different tasks, but two of them are low resource and two of them translate into German, which was necessary because we only had access to German language experts for preparing the evaluation.

References


A Appendix

A.1 Pronoun and Formality Translation Evaluation

Here, we explain how we calculate the pronoun translation and formality translation F1 scores.

Pronouns

For each triplet $(F_n, E_n, \hat{E}_n)$ (source, hypothesis, reference) of our test data we first check if it contains a valid ambiguous pronoun. That means, in the source sentence there must be an English 3rd person pronoun in the neutral form and it also must be labeled as a pronoun by the English POS-tagger. We also check if a 2nd or 3rd person plural pronoun is present in the source and if that is the case, we do not consider female pronouns on the target side, since we could not distinguish if e.g. ‘sie’ is the translation of ‘it’ or ‘they’. This would require a word alignment between source and hypothesis/reference which we do not have. If we found the example to be valid, we then check for occurrences of 3rd person pronouns in the male, female and neuter forms, in both reference and hypothesis using a German POS-tagger as well as language-specific regular expressions. After going through the complete test data $(F_n, E_n, \hat{E}_n)$ sentence-by-sentence we calculate an F1 score for pronoun translation:

$$F1_{pro} = \frac{2 \cdot P_{pro} \cdot R_{pro}}{P_{pro} + R_{pro}}$$

with precision $P_{pro} =$

$$\frac{\sum_{n=1}^{N} \sum_{x} \min \left( CP(F_{n}, E_{n}, x) , CP(F_{n}, \hat{E}_{n}, x) \right)}{\sum_{n} \sum_{x} CP(F_{n}, E_{n}, x)}$$

and recall $R_{pro} =$

$$\frac{\sum_{n=1}^{N} \sum_{x} \min \left( CP(F_{n}, E_{n}, x) , CP(F_{n}, \hat{E}_{n}, x) \right)}{\sum_{n} \sum_{x} CP(F_{n}, E_{n}, x)}$$

where $CP(\cdot, \cdot, \cdot)$ counts the number of valid pronoun occurrences and $x \in \{\text{male}, \text{female}, \text{neuter}\}$.

Formality

We follow almost exactly the same steps as for detecting the pronoun translations described above. The only differences are that we check for validity slightly differently and instead of pronouns we check for occurrences of formal/informal style. For sentence-pairs where 3rd person female/neuter or 3rd person plural pronouns are present, we do not count the formal occurrences, since we might not be able to distinguish the German translations in these cases. We calculate an F1 score for formality translation using

$$F1_{for} = \frac{2 \cdot P_{for} \cdot R_{for}}{P_{for} + R_{for}}$$

with precision $P_{for} =$

$$\frac{\sum_{n=1}^{N} \sum_{x} \min \left( CP(F_{n}, E_{n}, x) , CP(F_{n}, \hat{E}_{n}, x) \right)}{\sum_{n} \sum_{x} CP(F_{n}, E_{n}, x)}$$

and recall $R_{for} =$

$$\frac{\sum_{n=1}^{N} \sum_{x} \min \left( CP(F_{n}, E_{n}, x) , CP(F_{n}, \hat{E}_{n}, x) \right)}{\sum_{n} \sum_{x} CP(F_{n}, E_{n}, x)}$$

where $CP(\cdot, \cdot, \cdot)$ counts the number of valid pronoun occurrences and $x \in \{\text{formal}, \text{informal}\}$.

The POS-tagger we use are en_core_web_sm\(^2\) for English and de_core_news_sm\(^3\) for German. For both languages, spaCy claims an accuracy of 97% for POS-tagging and in our testing we did not find even a single error in pronoun-tagging. For calculating the Pronoun Translation F1 score we use the same ContraPro test set as described in Section 3.5 with the correct references. For calculating the Formality Translation F1 score, we use the test set from the OS En-De task. The statistics for both test sets are reported in Table 8. In the ContraPro test set, for each gender class we have exactly 4,000 examples. The fact that we identify more than 4,000 valid examples for the pronoun case means, that in some cases we identify multiple pronouns per sentence. All in all, we find the classes to be relatively balanced for these test sets.

A.2 Dataset Statistics and Experimental Setups

For the NEWS En→De task, the parallel training data comes from the NewsCommentary\(^4\) corpus. As validation/test set we use the WMT newstest2015/newstest2018 test sets from the WMT news translation tasks (Farhad et al., 2021). For the TED En→It task, the parallel training data comes from the IWSLT17 Multilingual Task (Cettolo et al., 2017). As validation set we

\(^2\)https://spacy.io/models/en
\(^3\)https://spacy.io/models/de
\(^4\)https://data.statmt.org/news-commentary/v14/
use the concatenation of IWSLT17.TED.dev2010 and IWSLT17.TED.tst2010 as test set. For the OS En→De task, the parallel training data comes from the OpenSubtitlesV2018 corpus (Lison et al., 2018). We use the same train/validation/test splits as Huo et al. (2020) and additionally remove all segments that are used in the ContraPro test suite (Müller et al., 2018) from the training data. The data statistics for all tasks can be found in Table 9.

<table>
<thead>
<tr>
<th>task</th>
<th>dataset</th>
<th># sent.</th>
<th># doc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWS</td>
<td>train</td>
<td>330k</td>
<td>8.5k</td>
</tr>
<tr>
<td></td>
<td>valid</td>
<td>2.2k</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>3k</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>ContraPro</td>
<td>12k</td>
<td>12k</td>
</tr>
<tr>
<td>TED</td>
<td>train</td>
<td>232k</td>
<td>1.9k</td>
</tr>
<tr>
<td></td>
<td>valid</td>
<td>2.5k</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>1.1k</td>
<td>10</td>
</tr>
<tr>
<td>OS</td>
<td>train</td>
<td>22.5M</td>
<td>29.9k</td>
</tr>
<tr>
<td></td>
<td>valid</td>
<td>3.5k</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>3.8k</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>ContraPro</td>
<td>12k</td>
<td>12k</td>
</tr>
</tbody>
</table>

Table 9: Data statistics for the different document-level translation tasks.

Since in the original release of ContraPro only left side context is provided, we extract the right side context ourselves from the OpenSubtitlesV2018 corpus based on the metadata information of the segments. For translation of the ContraPro test set, as well as for scoring the contrastive references, we take both the left- and the right-side context into account. For the full-document systems, we cap the context size for the ContraPro test set to 4 sentences for computational reasons.

We tokenize the data using byte-pair-encoding (Sennrich et al., 2016; Kudo, 2018) with 15k joint merge operations (32k for OS En→De). The models are implemented using the fairseq toolkit (Ott et al., 2019) following the transformer base architecture (Vaswani et al., 2017) with dropout 0.3 and label-smoothing 0.2 for NEWS En→De and TED En→It and dropout 0.1 and label-smoothing 0.1 for OS En→De. This resulted in models with ca. 51M parameters for NEWS and TED and ca. 60M parameters for OS for both the sentence-level and the document-level systems.

Let us assume that the training data $C$ consists of $M$ documents $D_m$ and each document consists of source-target sentence pairs $(F_{n,m}, E_{n,m})$. The goal of training is to find the optimal model parameters $\hat{\theta}$ which minimize the loss function:

$$\hat{\theta} = \arg \min_{\theta} L(\theta)$$

When training the local context models, we define the loss function:

$$L(\theta) = -\frac{1}{|C|} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \log p_{\theta}(E_{n,m}^{n,m} | F_{n-m,k,m}^{n,m})$$

When we take full documents as input to the model, the loss function simply becomes

$$L(\theta) = -\frac{1}{M} \sum_{m=1}^{M} \log p_{\theta}(E_{1,m}^{N_m,m} | F_{1,m}^{N_m,m})$$

All systems are trained until the validation perplexity does no longer improve and the best checkpoint is selected using validation perplexity as well. Training took around 24h for NEWS and TED and around 96h for OS on a single NVIDIA GeForce RTX 2080 Ti graphics card. Due to computational limitations, we report results only for a single run.

For the generation of segments (see Section 3.4), we use beam-search on the token level with beam-size 12 and length normalization.
Unpacking Ambiguous Structure: A Dataset for Ambiguous Implicit Discourse Relations for English and Egyptian Arabic

Ahmed Ruby¹  Sara Stymne¹  Christian Hardmeier²

¹Uppsala University, Department of Linguistics and Philology
²IT University of Copenhagen, Department of Computer Science
{ahmed.ruby, sara.stymne}@lingfil.uu.se, chrha@itu.dk

Abstract

In this paper, we present principles of constructing and resolving ambiguity in implicit discourse relations. Following these principles, we created a dataset in both English and Egyptian Arabic that controls for semantic disambiguation, enabling the investigation of prosodic features in future work. In these datasets, examples are two-part sentences with an implicit discourse relation that can be ambiguously read as either causal or concessive, paired with two different preceding context sentences forcing either the causal or the concessive reading. We also validated both datasets by humans and language models (LMs) to study whether context can help humans or LMs resolve ambiguities of implicit relations and identify the intended relation. As a result, this task posed no difficulty for humans, but proved challenging for BERT/CamelBERT and ELECTRA/AraELECTRA models.

1 Introduction

Coherence is essential for effective communication in written or spoken language (Adornetti, 2015), and discourse connectives play a crucial role in achieving it by helping readers or listeners to infer the intended discourse relation holding between two text spans (Asr and Demberg, 2020). Listeners generally have little difficulty recovering the intended meanings with implicit connectives which are inferred between two juxtaposed independent sentences. They evidence this by combining lexical cues, general reasoning, and prosodic cues to effectively identify the implicit discourse relation. When interpreting ambiguous implicit relations, prosodic cues can be used for disambiguation in spoken language (Tyler, 2014; Jasinskaja, 2009), while semantics is essential in both speech and writing, ensuring effective communication and understanding. Consider for instance the following examples:

(a) John is tall, so she will ask him out.
(b) John is tall, but she will ask him out.
(c) John is tall. She will ask him out.

The discourse relations in both (a) and (b) can be understood by listeners and readers because the connectives "so" and "but", respectively, explicitly indicate the discourse relation. Although the implicit discourse relation is ambiguous in (c), listeners might be able to infer it through prosody. However, it is still an open question whether specific prosodic cues are helpful for disambiguation in this case. Moreover, disambiguation can also be achieved in written and spoken language using semantic cues (e.g., additional context), such as adding different preceding context sentences that can enforce either the causal or the concessive reading. For instance, the preceding context for the casual and concessive reading can be:

(1) She prefers tall men. John is tall. She will ask him out.
(2) She prefers short men. John is tall. She will ask him out.

The additional context can influence the ambiguous structure, suggesting a likely interpretation in (1) that her preference for tall men implies a causal relation, while her preference for short men indicates a concessive interpretation in (2).

We observed that the ambiguous structure of implicit relations arises when the first argument (Arg1) does not provide specific details about the event being described and can be influenced by additional context information. However, for some ambiguous examples or structures, there is a clearly preferred reading even without any context, unless there is extremely strong evidence for a different reading. Consider for instance the following example (adapted from (Carston, 1993))

(a) Max fell. John pushed him.
The preferred reading for this example is that the pushing caused the falling. However, there is another possible reading where Max fell first and was later pushed by John, but it needs extremely strong evidence to force this interpretation. This means that it is hard to figure out if other aspects than semantics contribute to inferring the intended reading.

In order to explore how ambiguous implicit relations can be successfully resolved by the listener, we plan to conduct, in future work, a controlled experiment on the impact of prosody without being disambiguated by the semantic component. To support this, this study presents a small dataset of "truly" ambiguous examples for implicit discourse relations for both English and Egyptian Arabic, which cannot be resolved in the absence of any context, so that it enables a future investigation of prosodic features. We create a set of sentences with an implicit discourse relation that can be ambiguously read as either causal or concessive with two different preceding context sentences forcing either the causal or the concessive reading. The dataset is validated by humans who read these sentences and filled in the intended implicit discourse connective by choosing the most appropriate option from the provided list of connectives.

We were able to identify the ambiguous structure of implicit discourse relations and propose a new set of principles to construct ambiguity in implicit discourse relations. This process led to the creation of a small dataset for English and Egyptian Arabic that was validated by human participants. As far as we are aware, this is the first dataset that addresses ambiguous implicit discourse relations.

Since human participants were able to identify the intended implicit connectives in a set of examples, we investigate whether language models like BERT (Devlin et al., 2019) and ELECTRA (Clark et al., 2020) can also fill in the implicit connectives in the examples correctly, which is a challenging task, as context barely influenced the choice made by these models.

2 Related Work

2.1 Discourse relation datasets

Although discourse relations have been extensively studied over the last two decades, leading to elaborate taxonomies and inventories of varying scope and levels of abstraction, it is still challenging to provide a general definition for implicit discourse relations (Jasinskaja, 2009). However, there are some inferred relation types that are considered in Wolf and Gibson (2004); Miltsakaki et al. (2005); Prasad et al. (2008); Lavid and Hovy (2010) and annotated implicit relations were covered in the Penn Discourse Tree Bank 2.0 (PDTB 2.0) (Prasad et al., 2008), which is the most popular resource. Moreover, there are discourse-annotated corpora that cover implicit relations in multiple languages, such as TED Multilingual Discourse Bank, or TED-MDB, which contains transcribed TED talks in English, German, Russian, European Portuguese, Polish, and Turkish (Zeyrek et al., 2020), as well as in individual languages following the PDTB approach, such as the Hindi Discourse Relation Bank (Oza et al., 2009) and the Chinese Discourse Tree-banks for Chinese. (Yuping et al., 2014; Long et al., 2020).

2.2 Discourse relations and ambiguity

Ambiguous structures can signal multiple potential interpretations of implicit discourse relations, and the intended relation can be inferred by context or by drawing on one's background assumption (Verhagen, 2000). Our study focuses on ambiguous implicit discourse relations, where a two-part sentence implies various potential relations, and must be inferred by context. Considering the distribution of discourse connectives in both PDTB and LADTB as reported by (Alsaif, 2012), the connectives 'but' and 'so' are commonly used in English and Arabic (Pitler et al., 2008; Alsaif, 2012). This observation has inspired the present study to explore the implicit relations that can be expressed by these particular connectives.

3 Ambiguity in inferring implicit relations

Each discourse relation involves two arguments, which are typically expressed as two clauses or phrases (Cabrio et al., 2013). Muskens (2000) argues that underspecified representations must be ambiguous. Drawing from this notion, we have shaped our own study’s approach to examining the first argument with ambiguity in mind. The results of the validation confirm that if Argument 1 does not provide information that is relevant to inferring a specific discourse relation, it is not possible to make an inference about that relation unless there are underlying assumptions or presuppositions. In this case, it may be necessary to look for additional
information from context to infer the implicit discourse relation.

The meaning of Argument 1 can be shaped and influenced by context if it carries a neutral connotation, and Argument 2 gives additional information or detail based on the event influenced by context. Consider the example in Figure 1, where Argument 1 "John is tall" in both sentences is unspecified and needs to be interpreted in the context of the sentence to determine the intended information conveyed by Argument 2 "she will ask him out". In the first sentence, the context helped Argument 1 convey a positive meaning to infer that she has a preference for tall men, and because John fits this preference, she will ask him out, while in the second sentence, the context helped Argument 1 convey a negative meaning to infer that she does not have a preference for tall men, but she will still ask John out, even though he fits this preference.

While there is a lot of evidence that the context can disambiguate the discourse relation structure (Nowak and Michaelson, 2020; Lichao, 2010), we still do not have a thorough understanding of how ambiguity in implicit relations is structured, and how they can be interpreted only by context. In this regard, this study examines whether different preceding context influence whether the causal or concessive reading is elicited. A dataset was created and validated to investigate this question in two languages (English and Egyptian Arabic).

4 Constructing data for ambiguous implicit discourse relations

Creating a dataset for implicit discourse relations that involve ambiguity can be a challenging task. This is because the ambiguous structure is not linguistically defined in a way that influences meaning. Furthermore, inferring implicit discourse relations can be difficult, since it requires a nuanced understanding of language and discourse. Therefore, we aim to investigate this gap by identifying the ambiguous structure of implicit discourse relations and proposing a method to build a dataset for inferring relations by context.

4.1 Principles of constructing ambiguity in discourse relations

The initial validation findings, which are detailed in Section 4.4, reveal several principles that can be used when constructing ambiguity in implicit discourse relations, such as:

1. The discourse relation between sentences or Arg 1 and Arg 2 should be implicit, where:
   (a) Arg 1 and Arg 2 are not connected by any structural connective, such as "so", "but", "because", etc., but the connective can still be inferred.
   (b) Arg 2 should not contain a lexical item e.g., "this" or "that" which implies a presupposition already established by Arg 1. This is because these anaphoric pronouns refer to the fact expressed by the first sentence, so the second sentence presents an evaluation of that fact, due to the lexical semantics of the verb that follows these pronouns. (Jasinskaja, 2009)
   (c) Arg 2 should provide supplementary information or clarification for Arg 1.

2. Arg 1 should convey a neutral meaning\(^1\), and be influenced by context. For instance, "The apple is red" can be influenced and shaped by context to be positive or negative.

3. The discourse relation can only be inferred by context.

4.2 Data and design

We create a set of contrastive sentences pairs that deliberately contain discourse relation ambiguities, with their preceding context, where both Arg1 and Arg2 are identical, with "but" versus "so", depending on if the context makes Arg 2 expected or unexpected. As shown in Table 1.

<table>
<thead>
<tr>
<th>Context</th>
<th>Target_sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>The car is very cheap</td>
<td>It's 100,000, ___ I'll buy it.</td>
</tr>
<tr>
<td>The car is very expensive</td>
<td>It's 100,000, ___ T'f I'll buy it.</td>
</tr>
</tbody>
</table>

Table 1: Paired example with ambiguous implicit relations with two different preceding context sentences forcing either the causal or the concessive reading.

As you can see in Table 1, the preceding context sentence guides the speaker to the intended meaning of the target sentence whether the discourse relation between adjacent sentence is a causal or concessive relation, and thus the speaker can fill in the connective in these sentences depending on the context.

\(^1\)Arg 1 can also contain contronyms, which have opposite or contradictory meanings such as "crazy prices" can have opposite meanings depending on the context, it could refer to the low prices, or it could refer to the high prices. However, when testing this principle, we realized that it may be inferred by drawing on one’s background assumption.
She prefers tall men. John is tall. [so] she will ask him out.

She prefers short men. John is tall. [but] she will ask him out.

Figure 1: An example of inferring by context of the causal and concessive relation.

4.2.1 Arabic translation

There are five levels of Arabic used in Egypt as stated by Badawi (1973) in his socio-linguistic analysis of contemporary Arabic in Egypt: 1) Classical Arabic of the heritage, 2) Modern Standard Arabic, 3) Colloquial of the educated, 4) Colloquial of the enlightened, and 5) Colloquial of the illiterate. The "Colloquial of the educated" is a form of Arabic spoken by educated individuals that balance regional informality and linguistic proficiency. We opted for this level in our study, as opposed to Modern Standard Arabic (MSA) or "Colloquial of the enlightened," because it represents the prevalent form of spontaneous spoken communication. While MSA is widely understood, it is mainly used in formal written contexts or speeches, whereas "Colloquial of the enlightened" is characterized by its localized nature, which might limit understanding across regions or social groups.

By choosing the "Colloquial of the educated" level, we translated our English examples into Egyptian Arabic. In order to ensure consistency with the common writing style in Egyptian Arabic, two linguists, who are native Arabic-speaking, were asked to provide their feedback and suggestions about the writing style of the examples. This process helped to enhance the quality of the translation. After the first round of validation on translated examples, we decided to eliminate certain examples and introduce new ones. This implies that the English data is not entirely equivalent to the Arabic data.

4.3 Data validation method

In order to examine our data, we utilize human validation with the aim of ensuring the reliability and confidence of examples. This involves a number of procedures:

4.3.1 Distractors

To distract the respondents from the purpose of the study, and reinforce the impression that participants were reading the sentences naturally, we randomly interleave a number of distractors/fillers with the target examples, which reflect the other implicit discourse relations: expansion and temporal according to the PDTB relations hierarchy (Prasad et al., 2008). Since distractors should be fitted syntactically in all examples, we created 5 examples with implicit 'in fact' connective for expansion relation, and also 5 examples with implicit 'when' connective for temporal relation. These distracted examples are similar to the target examples in terms of design and construction, where contain two discourse units, e.g. clauses or sentences, with proceeding context such as:

(a) Writing on walls is illegal. The teacher arrived early in the morning, _ we were painting on the wall.

(b) Many people were thankful for the experience of traveling by car to Sharm El-Sheikh. We tried to travel there by car, _ it was a very wonderful experience.

For Arabic, we used the same procedures as those applied in English, but we found that the equivalent of the "when" connective, lámma, can convey both causal and synchronous relations simultaneously, which means that this equivalent can be fitted in with both relations. As a result, we decided to eliminate it and use the "at the time/sāriðthā" connective instead for the temporal relation.
4.3.2 Questionnaire design

To achieve our aim of distraction, we added additional connectives, resulting in a balanced distribution of fillers and actual test connectives. As a result, the selection list comprises four connectives: "when", "so", "but", and "in fact", as illustrated in Figure 2.

Figure 2: selection list

Regarding Arabic, there are more equivalent words for these connectives in Egyptian Arabic, such as "so" has three equivalents, and "but" has two equivalents as shown below.

so
fa yašān-kida
fa-yašān-kida
but
bass läkin

For the "so" connective, (fa) the first equivalent can also be used as a filler, so we exclude it in this experiment, as it can be fitted in with other connectives, and the second and the third equivalent are similar in use, but the second is more widely used. We decided to use the second equivalent yašān-kida in the experiment.

The first equivalent of the "but/bass" connective is more commonly used, but the results of the first iteration showed that it can also be used as a filler, whereas the second one is mainly used in Modern Standard Arabic and among the educated in colloquial language. Therefore, we decided to use "läkin" as the equivalent of "but" connective in the second iteration.

We also found that the "in fact" connective has two equivalents, the first one is fil-hqīqa, which can be also used as a filler, and the second one is bil-fīšl, which is more commonly used among the educated in colloquial language. We ultimately decided to use the latter.

We use the same ratio of test items and fillers as that used in the English experiment in both validation iterations. In the first iteration, we used the Arabic equivalents of distractors/fillers used in the English experiment: "when/lānma", "so/yašān-kida", "but/bass" and "in fact/fil-hqīqa". While in the second iteration, we replaced the "when" connective with "at the time/sāyiθā" and used another Arabic equivalent of "in fact" connective and "but" connective: "at the time/sāyiθā", "so/yašān-kida", "but/lākin" and "in fact/bil-fīšl".

4.3.3 Participants

In order to investigate whether humans are able to identify implicit discourse connectives for these examples, we designed a questionnaire in English and invited volunteers with diverse native languages to answer the questionnaire by selecting the most appropriate connective from the provided list of options to fill in the blanks. In the first and second iterations, 24 and 21 participated in this validation, respectively.

We used the same process to create a questionnaire in Egyptian Arabic and invited Egyptian Arabic speakers to answer the questionnaire by selecting the most appropriate connective from the provided list of options to fill in the blanks as well. In the first and second iterations, 19 and 28 native speakers of Egyptian Arabic participated, respectively.

4.3.4 Procedure

To perform the validation process, we utilized the SurveyMonkey platform and enabled the randomization feature to randomize the two context sentences across participants so that each participant will only see one variant of each example. We distributed the survey link via an email list to gather responses from volunteers. In this task, we marked the main sentence in boldface, which contained the missing connective and preceded by context, and added a list of connectives under each sentence, as illustrated in Figure 2. The task was organized into three blocks of questions and was followed by a few language-related questions presented in Appendix C. However, these questions were not used for analysis. The entire validation task took approximately 10 minutes to complete.

The validation of the Egyptian Arabic dataset was also run by using SurveyMonkey. To collect responses from Egyptian people, we used Facebook to distribute the survey link and request their participation in answering the questions. Following the same processes of the English validation, where
each participant can only see one variant of each example and is not allowed to do the validation twice.

### 4.4 Results and Analysis

This section presents the summary of validation results on both English and Egyptian Arabic examples, showing the results of the So/But grouping that was conducted to compare the performance of the pair examples. Two validation iterations were conducted in both English and Egyptian Arabic. After analyzing problematic cases and refining our principles from the first English iteration, we created significantly improved examples for the second English iteration. Similarly, by examining the results of the first Egyptian Arabic iteration and adjusting the corresponding connective words in Egyptian Arabic, we achieved much better outcomes in the second Egyptian Arabic iteration. To ensure the validity and reliability of the validation process, a minimum threshold of 80% agreement between participants was established, meaning that only paired examples with a high level of agreement were included. Table 2 shows the summary of the validation results for both languages within each iteration. Detailed validation results can be found in Appendix A. We also employ Krippendorff’s alpha to determine the degree of agreement or reliability among annotators/participants for each variant of an example. More detailed results of the inter-annotator agreement can be found in Appendix B.

In the first iteration of the English validation, which was performed on 31 paired examples, only one example in both cases: causal and concession met the threshold, while 14 examples in both cases did not meet the threshold. However, in the second iteration, which involved 32 paired examples, the findings reveal that the participants were relatively successful in selecting the appropriate connective, with 22 examples in both cases meeting the threshold and only one example in both cases failing to meet the threshold.

To maintain consistency, we followed our principles and added three more examples to the existing set in English, so that both datasets contain a total of 22 examples.

For our preliminary dataset, the findings reveal that the participants were relatively successful in selecting the appropriate connective in the ‘so’ group. However, the results for the ‘but’ group were less promising, indicating that the participants struggled to identify the correct connective in these instances. This could be attributed to a range of factors, such as difficulties in understanding the intended meaning, or Arg 1 carries underlying assumptions or presuppositions. For instance, instead of considering the contextual cues in the examples below, several participants relied on their presuppositions about how to interpret the meaning of Arg 1. Here is the phrase “the weather changed” that can carry presuppositions as it can be to the better or to the worse:

(a) One day it was nice and sunny so my family and I decided to go on a trip. Suddenly the weather changed, [...] we decided not to go.

(b) On the morning of the game, it was cloudy and rainy. Suddenly the weather changed, [...] we decided not to go.

Here is also the phrase "The time was short" typically implies a negative outcome rather than a positive one, especially when the second argument indicates the result of the event:

(a) The guest lecturer we had this week was much less long-winded than our usual professor. The time was short, [...] I had fun.
(1) The car is very cheap. It’s 100,000, [MASK] I’ll buy it.
The car is very expensive. It’s 100,000, [MASK] I’ll buy it.
(2) He prefers long movies. The movie was only 30 minutes, [MASK] he
did not like it.
He prefers short movies. The movie was only 30 minutes, [MASK] he
did not like it.
(3) She likes eating green apples. The apple was green, [MASK] She ate it.
She likes red apples. The apple was green, [MASK] She ate it.
(4) I really like mint. This tea had mint in it, [MASK] I drank it.
I do not like mint. This tea had mint in it, [MASK] I drank it.

(b) I spent a great time with my family. The
time was short, [...] I had fun.

Consequently, we removed instances with con-
fusing or inconsistent results, modified some
phrases to improve clarity, and added new instances.
These changes led to a second iteration of human
validation.

5 Pilot experiment

Since human participants were able to identify the
intended implicit connectives in a set of examples,
we now investigate whether language models like
BERT and ELECTRA will also be able to correctly fill in the implicit connectives within the provided examples.

5.1 English version
For English examples, we use the uncased version of the bert-base and electra-base models from Hugging Face\(^2\) by inserting a mask between Arg 1 and Arg 2 to fill in the missing word, with setting up the topk parameter to 3 to obtain the top 3 predicted words.

5.2 Arabic version
We use CAMeLBERT-Mix (bert-base-arabic-camelbert-mix) model (Inoue et al., 2021) from Hugging Face as well, which is trained on a mixture of Modern Standard Arabic (MSA), Dialectal Arabic (DA) and classical Arabic (CA) variants, to fill in the implicit words for Arabic examples by inserting a mask between Arg 1 and Arg 2, with setting up the topk parameter to 3 to obtain the top 3 predicted words. We also use AraELECTRA-base-generator (Antoun et al., 2021) from Hugging Face, with the same setting.

5.3 Results and Analysis
The outcomes for the top three predicted words by BERT and ELECTRA on paired English examples are detailed in Appendix D. Figure 3 displays here the top predictions and their corresponding scores from BERT for 4 paired English examples with masked connectives.

These results indicate that identifying implicit discourse connectives is quite challenging for language models due to not capturing the influence of context on Arg 1 as there are small differences in the predictions for both So/But groups.

The results of the top three predicted words for Arabic examples, which encompass 25 and 33 different words in BERT and ELECTRA respectively, are also illustrated in Appendix D. Figure 4 presents here the top predictions and their corresponding scores from BERT for 4 paired Egyptian Arabic examples that include masked connectives. These examples are translations of the examples in Figure 3, following the same order.

These results indicate that identifying implicit discourse connectives for Arabic examples is quite challenging as well, as context barely influenced the choice made by these models. Furthermore, the performance of the models on Arabic examples is extremely poor, as many of the predicted words do not function as connectives. As shown in the legend entries of the figures, the words enclosed in black squares are connectives, while others are not. This can be interpreted for several reasons:

1. There are potentially systematic differences in the prevalence of implicit discourse relations in spoken data compared to written texts (Rehbein et al., 2016).

2. A discourse relation can be communicated by a pair of clauses conjoined by "and", but the sentences are not connected asyntactically (Jasinskaja, 2009; Rohde et al., 2018). For example, the Result relation can be communicated implicitly both with or without and, such as (Jasinskaja, 2009):
   
   (a) She fed him poisoned stew and so he died.
   (b) She fed him poisoned stew and he died.
   (c) She fed him poisoned stew. He died.

   The connective "so" in (a) explicitly indicates a causal connection, but the same relation is successfully conveyed in (b) and (c), despite the absence of "so" or even "and".

   This can explain the appearance of "and" in both the legend entries of English and Arabic results. Since "wa/and" is proclitic in Arabic, which is usually attached to the word (Habash, 2010), it may provide an explanation for the appearance of the words enclosed in red squares within legend entries of the Arabic results.

6 Conclusion and Future Work
In this paper, we introduced principles of constructing and inferring ambiguity in implicit discourse relations, and created a dataset for ambiguous implicit discourse relations, specifically causal and concessive relations for both English and Egyptian Arabic. We also validated both datasets by humans and language models (LMs) to study whether context can help humans or LMs resolve ambiguities of implicit relations and identify the intended relation. For future work, we plan to conduct a controlled experiment on the impact of prosody to figure out whether specific prosodic features correlate with the disambiguation of implicit discourse relations. We also intend to construct more examples to build a classification model to identify the two implicit discourse relations.

\(^2\)https://huggingface.co/bert-base-uncased

133
Acknowledgements

We extend our thanks to Hannah Rohde for her valuable insights, and to Rima Haddad for her feedback on our Arabic texts.

References


Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In ICLR.


A Human validation details

Figure 5 shows the results of the first iteration of human validation on English examples. This figure consists of two vertically stacked plots, each with four lines representing different categories: "When" (green stars), "so" (red circles), "but" (blue squares), and "in fact" (yellow triangles). The x-axis corresponds to the number of paired examples, labeled 1 to 31, while the y-axis represents the degree of agreement in responses. Each example has plotted points for each category.

In the first plot, the red "so" line has the highest values overall, with many points ranging from 80 to 100. The green "When" line remains at 0 for all data points. The blue "but" line has a few points above 10, but most of its points are at 0. The yellow "in fact" line is mostly below 20, with some points reaching above 20. On the other hand, the second plot shows the blue "but" line with the highest values, featuring many points ranging from 80 to 100. The red "so" line is mostly below 40. The green "When" line is mostly below 20, with some points reaching above 20 or 30. The yellow "in fact" line has a few points above 20, but most of its points are at or near 0.

The results of the second iteration of human validation on Egyptian Arabic examples are shown in Figure 6. In the first plot, the red "so" line has the highest values overall, with many points ranging from 80 to 100. The green "When" line remains at 0 for all data points. The blue "but" line has a few points above 10, but most of its points are at 0. The yellow "in fact" line is mostly below 20, with some points reaching above 20. On the other hand, the second plot shows the blue "but" line with the highest values, featuring many points ranging from 80 to 100. The red "so" line is mostly below 40. The green "When" line is mostly at 0. The yellow "in fact" line has a few points above 10, but most of its points are at 0.

There is a significant improvement in both the "so" and "but" groups. As a result, we decided to select the example pairs that scored above 80% and translate them into Egyptian Arabic for further validation.

Figure 7 shows the results of the first iteration of human validation on Arabic examples. In the first plot, the red "so" line has the highest values overall, with many points ranging from 80 to 100. The green "When" line has a few points, but most of its points are at 0. The blue "but" line has some points above 10, and the majority of its points are below 20. The yellow "in fact" line is mostly below 20, with a few points reaching above 20. On the other hand, the second plot shows the blue "but" line with the highest values overall, featuring many points ranging from 80 to 100. The red "so" line is mostly below 40. The green "When" line is mostly at 0. The yellow "in fact" line has a few points above 10, but most of its points are at 0.

The findings indicate that using some Arabic equivalents as fillers led to confusion, making it challenging for participants to identify the correct connective in these cases. Therefore, we tried to avoid using ambiguous equivalents and proposed alternative equivalents of the selection list. These changes also led to a second iteration of human validation.

The results of the second iteration of human validation on Egyptian Arabic examples are shown in Figure 8, indicating a significant improvement in both the "so" and "but" groups. In the first plot, the
Figure 5: The validation results of the first iteration on So/But groups of English examples

Figure 6: The validation results of the second iteration on So/But groups of English examples
Figure 7: The validation results of the first iteration on So/But groups of Egyptian Arabic examples

Figure 8: The validation results of the second iteration on So/But groups of Egyptian Arabic examples
red "so" line has the highest values overall, with many points ranging from 80 to 100. The green "When" line has a few points, but most of its points are at 0. The blue "but" line has some points above 10, and the majority of its points are below 20. The yellow "in fact" line is mostly below 20, with a few points reaching above 20. On the other hand, the second plot shows the blue "but" line with the highest values overall, featuring many points above 40 and the majority above 20. The red "so" line has a few points above 20, but it is mostly below 40. The green "When" line has a few points above 10, but most of its points are at 0. The yellow "in fact" line has a few points above 10, but most of its points are at 0.

As a result, we obtained 19 (to which we later added 3 more, totaling 22) and 22 examples of pairs scoring above 80% for English and Egyptian Arabic, respectively. Table 3 provides a summary of the final examples count for each discourse relation in each language.

### B Agreement Evaluation among Annotators

We use Krippendorff’s alpha, which is a statistical measure to determine the degree of agreement or reliability among annotators/participants, by calling krippendorff.alpha function from the krippendorff Python package. Since each variant was only scored by a subset of all participants, we calculate it separately for each variant of each question, based only on the choice given by the subset of participants.

Table 8 shows the evaluation of inter-rater reliability using Krippendorff’s Alpha calculation on the final English examples. It provides insights into the level of agreement among participants for each variant of an example, the concessive and causal relations. We observe that there are high agreement levels among the participants for most of the Causal and Concessive variants.

Table 9 illustrates the evaluation of inter-annotator agreement using Krippendorff’s Alpha calculation on the final Egyptian Arabic dataset. The findings also reveal a substantial degree of agreement among the participants for the majority of the Causal and Concessive variants.

### C Language-related questions

There were four language-related questions:

1. **What was the first language you learned as an infant?** Table 4 displays a summary of the responses.

2. **Were any other languages spoken by your cares at home before you were 6?** Table 5 provides a summary of the responses.

3. **Did you attend daycare where a different language was spoken before the age of 6?** Table 6 shows a summary of the responses.

4. **What other languages do you speak fluently?** Table 7 shows a summary of the responses.

### Table 3: Summary of the final examples for each discourse relation in each language

<table>
<thead>
<tr>
<th>Dataset</th>
<th>concessive</th>
<th>causal</th>
<th>concessive-causal pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>Arabic</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 4: The summary of answers for this question

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Validation</th>
<th>en</th>
<th>sv-SE</th>
<th>ar-EG</th>
<th>other</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>1st iteration</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2nd iteration</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Arabic</td>
<td>1st iteration</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2nd iteration</td>
<td>1</td>
<td>0</td>
<td>27</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5: The summary of answers for this question

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Validation</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>1st iteration</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>2nd iteration</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>Arabic</td>
<td>1st iteration</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>2nd iteration</td>
<td>3</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 6: The summary of answers for this question

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Validation</th>
<th>No</th>
<th>en</th>
<th>fr</th>
<th>other</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>1st iteration</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2nd iteration</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Arabic</td>
<td>1st iteration</td>
<td>9</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2nd iteration</td>
<td>13</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7: The summary of answers for this question
Table 8: Evaluation of Inter-Rater Reliability: Krippendorff’s Alpha Calculation on the Final English Dataset Using the Nominal Measurement Level

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sentence Pair No</th>
<th>Causal</th>
<th></th>
<th>Concessive</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of Participants</td>
<td>Agreement</td>
<td>No. of Participants</td>
<td>Agreement</td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>1</td>
<td>11</td>
<td>0.57</td>
<td>10</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12</td>
<td>0.67</td>
<td>9</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6</td>
<td>1.00</td>
<td>15</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>13</td>
<td>0.63</td>
<td>8</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>15</td>
<td>1.00</td>
<td>15</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>13</td>
<td>1.00</td>
<td>8</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>12</td>
<td>1.00</td>
<td>9</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>13</td>
<td>0.79</td>
<td>8</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>13</td>
<td>0.79</td>
<td>8</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>12</td>
<td>0.78</td>
<td>9</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>13</td>
<td>0.79</td>
<td>8</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>11</td>
<td>1.00</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>11</td>
<td>1.00</td>
<td>10</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>15</td>
<td>1.00</td>
<td>6</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>6</td>
<td>1.00</td>
<td>15</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>11</td>
<td>0.70</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>7</td>
<td>1.00</td>
<td>14</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>7</td>
<td>1.00</td>
<td>14</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>11</td>
<td>0.76</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>2</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>2</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>2</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 9: Evaluation of Inter-Rater Reliability: Krippendorff’s Alpha Calculation on the Final Egyptian Arabic Dataset Using the Nominal Measurement Level

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sentence Pair No</th>
<th>Causal</th>
<th></th>
<th>Concessive</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of Participants</td>
<td>Agreement</td>
<td>No. of Participants</td>
<td>Agreement</td>
<td></td>
</tr>
<tr>
<td>Arabic</td>
<td>1</td>
<td>8</td>
<td>0.67</td>
<td>7</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8</td>
<td>0.67</td>
<td>7</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>15</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>13</td>
<td>0.79</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>0.67</td>
<td>10</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>8</td>
<td>1.00</td>
<td>7</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>1.00</td>
<td>7</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>7</td>
<td>0.63</td>
<td>8</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>8</td>
<td>0.67</td>
<td>7</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8</td>
<td>0.67</td>
<td>7</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>8</td>
<td>0.67</td>
<td>7</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>9</td>
<td>0.71</td>
<td>6</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>6</td>
<td>1.00</td>
<td>9</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>7</td>
<td>1.00</td>
<td>8</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>7</td>
<td>0.63</td>
<td>8</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>6</td>
<td>1.00</td>
<td>9</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>7</td>
<td>0.63</td>
<td>8</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>8</td>
<td>0.67</td>
<td>7</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>8</td>
<td>1.00</td>
<td>7</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>4</td>
<td>1.00</td>
<td>11</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>10</td>
<td>1.00</td>
<td>5</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>3</td>
<td>1.00</td>
<td>3</td>
<td>1.00</td>
</tr>
</tbody>
</table>

D BERT and ELECTRA Validation

Figure 9 below shows the results of the top 3 predicted words on paired examples that scored above 80% in both cases in human validation for English. The figure presents the results of BERT on grouped examples through a pair of vertically aligned stacked bar charts. Each group represents the top predictions for masked connectives and their scores, which are the same in both groups (So, But, and And). In the first bar chart, the values of "so," "but," and "and" are distributed across the 22 bars/examples, with some bars showing a higher proportion of "so" or "but," and others displaying a higher proportion of "and." The second bar chart exhibits a similar distribution pattern. This means that identifying implicit discourse connectives is quite challenging for language models due to not
capturing the influence of context on Arg 1 as there are no differences in the predictions for both So/But groups.

The results of the top 3 predicted words for Arabic examples are illustrated in Figure 9. The figure presents the outcomes of BERT on grouped examples in Egyptian Arabic, utilizing a pair of vertically aligned stacked bar charts. Each group signifies the top three predictions for masked connectives along with their respective scores, which differ between the two groups. These predictions cover a total of 25 categories, with only 8 of them recognized as connectives.

In both plots, the highest values occur in categories 1, 4, and 19. Category 1 has the maximum value of 0.544, followed by category 4 with 0.482, and category 19 with 0.337. The results indicate that the model’s performance on Arabic examples is extremely poor since a considerable number of the predicted words do not function as connectives. In the Results and Analysis section, I presented some interpretations for these results.

Figure 10 shows the results of the top 3 predicted words, which are "so", "but", "and", "because" and "where", by ELECTRA on paired examples for English. The figure shows the results of ELECTRA on grouped examples through a pair of vertically aligned stacked bar charts as well. The plot reveals that the outcomes for ELECTRA didn’t differ much from the results of BERT. This observation further confirms that this task remains a substantial challenge for ELECTRA as well, primarily due to its limitations in capturing context.

The results of the top 3 predicted words for Arabic examples by AraELECTRA are illustrated in Figure 12. The figure shows the outcomes of AraELECTRA on grouped examples in Egyptian Arabic, utilizing a pair of vertically aligned stacked bar charts as well. These predictions cover a total of 33 categories, with only 11 of them recognized as connectives.
Figure 9: The validation results of BERT on So/But groups of English examples
Figure 10: The validation results of ELECTRA on So/But groups of English examples
Figure 11: The validation results of BERT on So/But groups of Egyptian Arabic examples
Figure 12: The validation results of AraELECTRA on So/But groups of Egyptian Arabic examples
Two-step Text Summarization for Long-form Biographical Narrative Genre

Avi Bleiweiss
BShalem Research
Sunnyvale, CA, USA
avibleiweiss@bshalem.onmicrosoft.com

Abstract
Transforming narrative structure to implicit discourse relations in long-form text has recently seen a mindset shift toward assessing generation consistency. To this extent, summarization of lengthy biographical discourse is of practical benefit to readers, as it helps them decide whether immersing for days or weeks in a bulky book turns a rewarding experience. Machine-generated summaries can reduce the cognitive load and the time spent by authors to write the summary. Nevertheless, summarization faces significant challenges of factual inconsistencies with respect to the inputs. In this paper, we explored a two-step summary generation aimed to retain source-summary faithfulness. Our method uses a graph representation to rank sentence saliency in each of the novel chapters, leading to distributing summary segments in distinct regions of the chapter. Bas- ing on the previously extracted sentences we produced an abstractive summary in a manner more computationally tractable for detecting inconsistent information. We conducted a series of quantitative analyses on a test set of four long biographical novels and showed to improve summarization quality in automatic evaluation over both single-tier settings and external baselines.

1 Introduction
Text summarization is a principal tool for reasoning about narrative structure and for understanding the content of a literary novel in a succinct form. Dated four decades back, the earlier seminal work by Lehnert (1981) pursued analytical summarization of narratives, and offered a graphical representation of human-generated plot units. In this graph, plot units are defined as conceptual elements referring to propositions or states that are linked by character relations. To produce a distilled version of the original discourse, a vast amount of information are selectively ignored by the reader. Similarly, traversing the graph identifies complex elements that are central to the story, and thus points of high relevance for summaries, and ones considered peripheral details.

e-summary: (1) It was committed in the presence of slaves, and they of course could neither institute a suit, nor testify against him; and thus the guilty perpetrator of one of the bloodiest and most foul murders goes unwhipped of justice, and uncensured by the community in which he lives. (2) He was, of all the overseers, the most dreaded by the slaves. (3) He was just proud enough to demand the most debasing homage of the slave, and quite servile enough to crouch, himself, at the feet of the master. [...] 

a-summary: The guilty perpetrator of one of the bloodiest and most foul murders goes unwhipped of justice, and uncensured by the community in which he lives. He was cruel enough to inflict the severest punishment, artful enough to descend to the lowest trickery.

Table 1: An example of text generation in our two-stage summarization. In the first step, we extract top-ranked sentences from an extended source chapter of a biographical novel with an average length of over 15K tokens. Then, we produce from the extracted summary (e-summary) an order-of-magnitude compressed abstractive summary (a-summary) that faithfully rephrases its predecessor. Shown are the leading three out of ten top ranked relevant sentences for the e-summary.

Recently, the domain of narrative understanding has gained interest of the research community (Piper et al., 2021). A wide array of computational models developed by language technology professionals provided for expressive generative textual-summaries. Presently, the prevailing approach to natural language generation (NLG) tasks, including summarization, is data driven and uses a sequence-to-sequence neural model pretrained on large text
corpora. Our work centers on evaluating the quality of producing summaries from chapters of long-form biographical novels. Unlike fictional narratives that require concatenating chapter summaries due to an inherit progressive plot nature, biographical chapters are relatively context independent and thus more readily manageable individually. Automatic generation of fluent summaries in the literary domain can be useful to complement the short description of a book provided by the author and to a certain extent assist in constructing expert critiques. The work by Berov (2019) demonstrated that a functional unit approach to summarizing computational storytelling can perform at around human level and contribute to better framing. We note that the narrative summarization task—while a rich source of innovation—is by and large untapped.

Pretrained language models based on the Transformer network (Vaswani et al., 2017) have achieved state-of-the-art performance generating fluent summaries from short input text. However, for long documents, model efficiency and summary quality characterized by remaining faithful to the respectful source present a challenge to natural language generation practitioners (Huang et al., 2021; Zhang et al., 2022). To mitigate the severity, NLG research applied both topical and generic approaches to the task of summary generation, distinguishing extractive summarization that produces high lexical overlap between a summary and the source document, and hence tends to be factually consistent. While abstractive summaries are prone to unaligned content that is not obviously inferable from the original text.

One of the more constraining facet of current neural models tasked with producing abstractive summaries is that the generated text can contain factually incorrect information with respect to the ground level text they are conditioned on. Summary inconsistencies are diverse and may include inversion, also known as negation, incorrect use of an entity that transpires as object swapping, or the introduction of an entity not in the original document, recognized as hallucination. Maynez et al. (2020) conducted a large-scale study and concluded that hallucination is the most critical to the coherence of abstractive summaries, while Cao et al. (2022) developed a detection approach that separates factual from non-factual hallucinations.

The complexity of the summarization task made automatic evaluation particularly challenging. In their recent line of work, Deng et al. (2021) proposed the intuition of information alignment between input and output text, and developed unified and interpretable metrics across a multitude of diverse NLG tasks. Distinctly for generative summaries, they offered effective definitions of relevance and consistency, widely identified as key aspects to characterize generation quality. Supported by robust theoretical grounds, their prevailing definitions strongly correlate with human judgment on how to concisely describe the most salient content in the input document. We adopted their interpretations in our empirical analysis and extended the consistency measure to a chapter-level rather than book-level over our test set of literary novels.

In Table 1, we present an overview of our two-step framework for summary generation. Distinguishing our work from prior research on extract-then-abstract methods, the approach we propose uses Transformer language models end-to-end, and experiments we conducted were run on exceptionally long-form chapters drawn from biographical literary novels. Our main contribution is twofold: (1) a high-quality and sustainable biographical literary dataset with each chapter consisting of its source text paired with both the extractive and abstractive summary constructs, and (2) through extensive experiments on a diverse biographical literary dataset, we demonstrate the effectiveness of our proposed approach and show similarity and consistency results that are exceeding or comparable to external baseline performance. Our biographical dataset is publicly accessible online.  

2 Related Work

We briefly survey existing methods that propose multi-stage text summarization systems evaluated on datasets from a broad range of domains.

Ling and Rush (2017) introduced a coarse-to-fine attention model that reads a document hierarchically, using coarse attention to select top-level blocks of text and fine attention to read the tokens of the chosen blocks. Their proposed summarizer scales linearly with the number of top-level chunks and effectively handles long sequences. However, their model performance lagged behind the standard instantiation baseline of the attention function on ROUGE similarity metrics.

Xu and Lapata (2020) proposed a coarse-to-fine modeling framework for extractive summarization.
applied to query focused multi-document. Their system incorporates a relevance estimator for retrieving textual segments—such as sentences or longer passages associated with a query—an evidence estimator which further isolates segments likely to contain answers to the query, and a centrali-

Our extractive summary component is reminiscent in spirit to their centrality estimator, however we use a Sentence Transformers (SBERT; Reimers and Gurevych, 2019) model to generate contextual sentence similarity embeddings that follows producing a sentence similarity matrix for computing graph centrality based ranking.

Pilault et al. (2020) explored Transformer language models and proposed an extract-then-summarize computational pipeline for long documents. Their model consists of an extractive element comprising a hierarchical neural encoder that outputs sentence representations of either a pointer to input sentences or to the result of sentence classification; and a Transformer language model conditioned on the extracted sentences as well as on either a part of or the entire input document to generate the summary. Their system was shown to outperform several baselines on similarity metrics, however, a discussion on factual correctness and consistency analyses of experimental results appears relatively sparse.

Gidiotis and Tsoumakas (2020) proposed a divide-and-conquer method by splitting the input into multiple segments, summarizing them separately, and combining the summary pieces. Basing on smaller source and target summary pairs that are focused on a specific aspect of the text, results in better alignment and considerable reduction of computation complexity. They used a basic sequence-to-sequence model and incorporated a rotational unit of memory (Dangovski et al., 2019) in its decoder that led to a more stable training and slightly improved F1 similarity scores. Content quality of their generated summaries relies entirely on ROUGE similarity metrics and could benefit from a broader evaluation framework such as offered by Deng et al. (2021).

More recently, Zhang et al. (2022) proposed a multi-stage split-then-summarize framework to generate summaries from long-form documents. Each source text divides into segments, matching each with a subset of target text. A coarse summary is generated for each segment and further concatenated as input to the next stage. After multiple stages of compression and summarization, a final stage produces a fine-grained summary. Their improved performance across baselines renders relatively low bi-gram scores, most likely owing to over-compression of source text.

An effective abstractive text summarization approach that first compresses long input text into a relatively short input sequence, and follows with efficient long-form document finetuning demonstrated comparable performance at a significantly lower computational cost (Choi et al., 2019; Su et al., 2020). Keen on a specific application, Pu et al. (2022) generate movie plots given movie scripts, by applying heuristic evaluation to extract actions and essential dialogues, a representation that reduces the average length of input movie scripts by 66%. Their system outperforms baselines on various automatic metrics.

3 Chapter Summarization

Our summarization task commences with producing an extractive summary from the source text of a book chapter, and follows with generating an abstractive summary from the salient extractive content (Table 1).

3.1 Importance Extraction

Extractive summarization generates text by selecting a subset of sentences in the original document. To this task we applied LexRank (Erkan and Radev, 2004) that computes sentence importance based on eigenvector centrality in a graph representation of sentences. The graph uses a cosine similarity matrix where each entry in the matrix is the similarity between the corresponding sentence pair. Formally, given \( n \) sentences in a novel chapter, we use a colon notation \( s_{1:n} = (s_1, \ldots, s_n) \) to denote the collection of sentences. We used bag-of-words to represent each sentence as a \( |V| \)-dimensional vector \( p \), where \( V \) is the chapter vocabulary. Hence, the similarity matrix \( M \in \mathbb{R}^{n \times n} \) contains elements \( m_{ij} = \text{sim}(p_i, p_j) \), where \( 1 \leq i, j \leq n \) and \( \text{sim} \) a similarity function. LexRank hypothesizes that sentences more similar to many other sentences in the book chapter are more central, or salient to the topic. The algorithm further emits the degree centrality of a node in the similarity graph—the count of similar sentences for each sentence.

Our extractive summarization task uses SBERT
SBERT derives semantically meaningful sentence embeddings that can be compared using cosine-similarity. We chose the distilled RoBERTa (Liu et al., 2019) variant of the BERT (Devlin et al., 2019) model, a pretrained Transformer network (Vaswani et al., 2017) on a paraphrase dataset. This model generates a dense embedding vector for each input sentence, of which we construct a similarity adjacency matrix $M$ that stores a weighted graph of all sentence-pairs. Matrix $M$ is further provided to LexRank for sentence importance ranking. The chapter extractive summary produced thus comprises a collection of top-ranked sentences with a sentence count that is proportional to the chapter text length, and commonly defaults to a defined maximal saliency.

3.2 Factual Abstraction

Extractive summary generation contrasts with abstractive summarization, where the information in the text is rephrased. Consistent with the Transformer architecture, BART (Lewis et al., 2020), considered a state-of-the-art model for the task of abstractive summarization, introduced denoising autoencoding objectives to pretrain sequence-to-sequence models. As a result, input texts are corrupted in two ways: (1) Text Infilling, where sampled token spans are replaced with a sequence of mask tokens [MASK], and (2) Sentence Permutation that splits a document into declarative sentences thereafter shuffled in random order.

Abstractive summary generation can be cast as a typical sequence-to-sequence learning problem. The pretraining objective of the core transformer model is to minimize the negative log-likelihood of the original document over corrupted text

$$
\mathcal{L}_G(\theta) = -\frac{1}{|Y|} \log p(Y|X; \theta),
$$

where $X$ is our extractive generated summary rendered as a set of sentences, $|Y|$ is the number of tokens in summary $Y$, and $\theta$ denotes the model parameters. In our experiments, we used the distilled version of BART, from which we drew sentence level representation for our automatic evaluation.

4 Information Alignment

The goal of a summarization task is to concisely describe the most salient information of the input text. Thus, the summary generated should be consistent and only contain content from the input, and the included content must be relevant. Using the intuition of information alignment, defined as the extent to which the information in one generative component is grounded in another, we can evaluate summary consistency and relevance (Deng et al., 2021).

More formally, let $x_{1:n}$ and $y_{1:m}$ be our respective extractive and abstractive summary text-sequences for each book chapter. Summary tokens are each represented with contextual embeddings we extracted from pretrained BERT (Devlin et al., 2019). Using embedding matching, the alignment vector $\text{align}(y \to x)$ consists of scores $\in [0,1]$ for each token in $y$, and amount to the maximum cosine similarity with the tokens in $x$

$$(i, j) = \arg\max_{i \in 1:n, j \in 1:m} \text{cossim}(x_i, y_j),$$

where $(i, j)$ is a pair of token indices pointing each to a distinct summary text sequence, and $1 \leq j \leq m$. The consistency metric that measures faithfulness thus follows naturally as the average of the alignment vector scores: $\text{mean}(\text{align}(y \to x))$. On the other hand, relevance is implicit in our two-step model that commences with ranking source sentences by their importance.

<table>
<thead>
<tr>
<th>Individual</th>
<th>Chapters</th>
<th>Tokens</th>
<th>FRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>11</td>
<td>154,293</td>
<td>77.5</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>60</td>
<td>620,312</td>
<td>75.1</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>70</td>
<td>1,269,660</td>
<td>65.3</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>115</td>
<td>2,238,248</td>
<td>65.5</td>
</tr>
</tbody>
</table>

Table 2: Metadata for our test set of biographical novels.

<table>
<thead>
<tr>
<th>Individual</th>
<th>Sentences</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>1,812</td>
<td>69</td>
<td>703</td>
<td>164.7</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>6,614</td>
<td>10</td>
<td>711</td>
<td>110.2</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>12,139</td>
<td>67</td>
<td>301</td>
<td>173.4</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>20,514</td>
<td>19</td>
<td>861</td>
<td>178.4</td>
</tr>
</tbody>
</table>

Table 3: Chapter sentence distribution across our test set of biographical novels.

5 Evaluation

Our proposed two-step summarization method is evaluated on our curated biographical literary test set. Automatic evaluation results are reported using
<table>
<thead>
<tr>
<th>Individual</th>
<th>e-summary</th>
<th>a-summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>14,989</td>
<td>1,835</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>98,059</td>
<td>9,458</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>121,865</td>
<td>12,206</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>248,878</td>
<td>20,209</td>
</tr>
</tbody>
</table>

Table 4: Token-length distribution of e-summary and a-summary across our biographical narrative test set.

the canonical ROUGE measure (Lin, 2004), and we have also experimented with the recently developed BARTScore metric (Yuan et al., 2021), more suitable to NLG tasks. We compared our performance with a handful of external baselines set to reach similar objectives like ours, and analyzed the newly proposed information alignment concept and consistency metric (Deng et al., 2021). Unless otherwise noted, we report novel-level summary quality using the average of chapter scores.

**Novel Test Set** We obtained unicode encoding of the literature text from Project Gutenberg, and carried our work on four biographies including Narrative of the Life of Frederick Douglass, An American Slave by Frederic Douglass (2006), Life on the Mississippi by Mark Twain (2004), Personal Memoirs of U. S. Grant by Ulysses S. Grant (2004), and Memoirs of Napoleon Bonaparte by Louis Antoine Fauvelet de Bourrienne (2006). These texts total 256 chapters and over four million words (Table 2). We also post for the literary set the Flesch Reading Ease (FRE) score that identifies a difficulty level range from standard to fairly easy.

In Table 3, we present chapter sentence distribution across our narrative literary set. Per book chapter there are on average 15,491 tokens (Table 2), and about 150 sentences with a little over 100 tokens per sentence. Chapter text is notably long in form and present a challenge to generate fluent and faithful summaries in a single computational pass.

**Generated Summaries** Our model provides two user-settable parameters to control summary generation: (1) the number of top-ranked sentences in a chapter ordered by their relevance to the input source text and concatenated to construct the e-summary. This number is set to ten by default; (2) the maximum token-length of the predicted abstractive summary set by the user to either fifty or one hundred words. We conducted ablation experiments and analyzed the impact of the bound token-length parameter on the a-summary generation quality. In Table 4, we provide token-length distribution of both e-summary and a-summary across our literary test set. On average, e-summaries consist of 387 tokens, while a-summaries, set to a maximum length of 50 tokens, have a mean of close to 40 words. Thus, the first stage of our summarization system presents a compression ratio of roughly 40 between source chapter text and e-summaries. In the second step, generated a-summaries are more concise than their respective e-summaries by an almost order of magnitude.

In Figure 1, we provide visualization of ten top-ranked sentences extracted from ten randomly sampled chapters in the Napoleon Bonaparte novel. We formulate extractive summaries as a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, where $m$ is the number of chapters in a book and $n$ the number of top-ranked sentences that are concatenated to found an extractive summary. In our setup, LexRank is set to return a fixed number of $n$ most relevant sentences, noting that the extracted list may contain ties. Over our experiments, we observed on average a fairly low— a slight over six percentage points— duplicated sentence salience across our test set. Most ties were an occurrence...
of two and a handful were of three sentences. Operating as a modular component, we applied the distilled RoBERTa-based pretrained SBERT model to generate contextual sentence embeddings. This model renders about 82 million trained parameters.

In our automatic evaluation we used the distilled checkpoint of BART, DistilBART-CNN-12-6, pretrained and finetuned on the CNN/Daily Mail news corpus (Nallapati et al., 2016) that comprises multi-sentence summaries, and on the extreme summarization dataset (XSUM; Narayan et al., 2018), both sustain a strong abstractive property. To generate a-summaries, we used the BART checkpoint model with a neural network of over 305 million parameters and ran inference on our biographical narrative test set.

**ROUGE Scores** We compute an a-summary from a reference e-summary. Rather than sentence-level that could potentially result in overlapping content and thus redundant summaries, we report summary-level ROUGE scores (Lin, 2004). Following standard practice, we chose F1 ROUGE as our evaluation metric to estimate the generation quality of summaries. Concretely, we used bi-gram ROUGE (ROUGE-2) that is a proxy for assessing informativeness and the longest common subsequence (ROUGE-L) to represent fluency. In Table 5, we show recall, precision, and F1 scores of produced a-summaries bound to a maximum token-length (maxlen) of 50 and 100 over our biographical literary set. Consistently ROUGE-L scores are higher than the respective bi-gram performance by about twenty five percentage points, on average. As expected, summary quality reduces proportionally to the e-summary token count (Table 4). Although limited to only two settings, our results support the conjecture that the longer the summary text sequence produced the higher the performance by up to 36%.

Table 5: ROUGE scores of a-summary generation (r - recall, p - precision, and f - F measure).

<table>
<thead>
<tr>
<th>Individual</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r</td>
<td>p</td>
<td>f</td>
<td>r</td>
</tr>
<tr>
<td>Frederick Douglass</td>
<td>0.13</td>
<td>0.95</td>
<td>0.23</td>
<td>0.18</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>0.11</td>
<td>0.91</td>
<td>0.19</td>
<td>0.16</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>0.10</td>
<td>0.92</td>
<td>0.18</td>
<td>0.15</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>0.08</td>
<td>0.91</td>
<td>0.15</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Table 6: ROUGE F1 scores for a single-tier setting. Summary maximum token-length is set to 500.

In Table 6, we report F1 ROUGE scores for a single-tier setting. This method collapses our summarizer stages and generates a-summary directly from the grounded source text of a chapter in a single computation pass. The summary maximum token length is implicitly set to 500 to account for the excessively long chapter document. Compared to our two-step summarization method, single-tier ROUGE-2 and ROUGE-L scores are shown to decline quadruply and triply, respectively.

We compared our summary generative performance with the quality of a half dozen of external baselines, presenting top F1 scores for both ROUGE-2 and ROUGE-L metrics in Table 7. At 0.29 F1, our ROUGE-2 measure exceeded state-of-the-art Gidiotis and Tsoumakas (2020) by 0.11 F1, while for ROUGE-L we came closely second with 0.38 F1 behind their best score of 0.41 F1. At an average of 15,491 words per novel chapter our dataset exceeded the token complexity of the baselines by at least 1.7X.

**BARTScore** We leveraged BARTScore (Yuan et al., 2021), a recently introduced evaluation metric for generated text that is unsupervised and does not require human judgments to train. Owing to its ability to utilize the entirety of the BART pretrained parameters, BARTScore can better support evaluation from a factual perspective. BARTScore relies on contextual word embeddings extracted

---

8 https://pypi.org/project/rouge/

9 We obtained code to calculate BARTScore from https://github.com/neulab/BARTScore
from pretrained sequence-to-sequence models and explores weighted conditional log-probabilities of a summary sequence given source tokens. In Table 8, we report BARTScore figures in average log-likelihood of probabilities $\in [0, 1]$. The calculated scores are less than zero, thus the higher the log-likelihood, the higher the probability. BARTScore appears far less affected by varying the maximum token-length of the produced a-summary, suggesting BARTScore captures aspects complementary to ROUGE. Consistent with ROUGE, BARTScore performance decreases with a higher e-summary word count.

<table>
<thead>
<tr>
<th>Individual</th>
<th>Pearson</th>
<th>Kendall</th>
<th>Spearman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>0.13</td>
<td>0.09</td>
<td>0.13</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>0.24</td>
<td>0.19</td>
<td><strong>0.27</strong></td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>0.23</td>
<td>0.17</td>
<td>0.24</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>0.14</td>
<td>0.11</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Table 9: BARTScore correlation between a-summary generation of 50 and 100 limited token-length.

Factual Consistency In this section, we offer qualitative analysis of factual consistency as it relates to biographical literary using embedding-matching alignment estimation. To extract contextual embeddings we used a pretrained BERT model that has nearly 109 million parameters. Our extractive summarization step warrants textually grounded generation of a summary, thus the following discussion pertains exclusively to the abstractive-summary computational stage. In Table 10, we show the distribution of factual correctness for aligning both (a-summary $\rightarrow$ e-summary) and (a-summary $\rightarrow$ source) across our biographical literary test set. The Frederick Douglass narrative scored the highest consistency of 0.81, along with the rest of the novels slightly behind, however, we contend that the three novels uphold a more faithful score of 0.77 owing to a larger sample of chapters. Using a comparable metric for compression tasks,

<table>
<thead>
<tr>
<th>Individual</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>0.41</td>
<td>0.75</td>
<td><strong>0.61</strong></td>
<td>0.11</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>0.62</td>
<td>0.92</td>
<td><strong>0.81</strong></td>
<td>0.11</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>0.42</td>
<td>0.89</td>
<td>0.63</td>
<td>0.10</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>0.48</td>
<td>0.89</td>
<td>0.67</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 10: Factual consistency distribution across our test set of biographical novels. The figures for each title show consistency measures for generated a-summaries, contrasting their alignment with the source text (grayed) and to their respective e-summary.
Deng et al. (2021) report consistency performance at 0.33 on the CNN/Daily Mail news corpus.

The impact on consistency performance gained by contrasting alignment of a-summaries with the source text and aligning a-summaries with e-summaries is a considerable 25% on average (Table 10). Evidently accurate automatic evaluation of generated summaries from long-form literary narratives is a multi-dimensional problem and pose a key challenge for optimization.

We note that extending the maximal generative token-length is not indefinite or else the summarizer aim to effectively balance both fluency and succinctness will be adversely affected.

<table>
<thead>
<tr>
<th>Individual</th>
<th>Finetuned Train</th>
<th>Finetuned Test</th>
<th>Pretrained F1</th>
<th>Pretrained F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>9</td>
<td>2</td>
<td>0.17</td>
<td><strong>0.30</strong></td>
</tr>
<tr>
<td>Mark Twain</td>
<td>48</td>
<td>12</td>
<td>0.14</td>
<td>0.26</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>56</td>
<td>14</td>
<td>0.17</td>
<td>0.27</td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>92</td>
<td>23</td>
<td>0.15</td>
<td>0.22</td>
</tr>
<tr>
<td>Unified</td>
<td>205</td>
<td>51</td>
<td>0.15</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 11: Contrasting ROUGE-L F1 scores for finetuned and pretrained BART models across our biographical novels. Finetuned narrative chapter allocations are shown for train and test subsets in individual and consolidated datasets.

**Finetuning** We explored finetuning the BART checkpoint on our biographical literary set and looked at the model ability to generalize across datasets. To this end, we built a distinct model for each and all novels unified, and applied an 80/20 percent chapter split for training and testing, respectively. We trained the BART model for three epochs using a cross-entropy loss, the Adam optimizer, a batch size of 32, and a learning rate of 1e-3. In Table 11, we present finetuned ROUGE-L F1 scores using a generation not to exceed a length of 50 tokens, and contrast them with the pretrained model (Table 5). Both finetuned and pretrained results follow a similar performance decline with a growing chapter token complexity. Finetuned scores are lower than the pretrained measures by about 1.75X on average, because the BART model weights are fitting to a much smaller dataset that is genre-different from the pretrained domain. Results of finetuning on the unified dataset appear commensurate with the rates obtained on individual novel data.

**Human Evaluation** Perceived as the best practice to evaluate auto-generated summaries, human judgment of long-form content similar to our scale remains challenging, time consuming, and often delivers only moderately reliable results. In a more recent study, Krishna et al. (2023) conducted a survey to understand best practices for applying human evaluation to summarization of large-scale documents. Their findings concluded that summaries derived from greater length articles are rarely evaluated by humans and the results obtained are often irreproducible.

<table>
<thead>
<tr>
<th>Individual</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederick Douglass</td>
<td>0.56</td>
<td>0.61</td>
</tr>
<tr>
<td>Mark Twain</td>
<td>0.63</td>
<td>0.67</td>
</tr>
<tr>
<td>Ulysses Grant</td>
<td>0.66</td>
<td><strong>0.72</strong></td>
</tr>
<tr>
<td>Napoleon Bonaparte</td>
<td>0.63</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 12: ROUGE F1 scores for a human evaluation. Summary maximum token-length is set to 100.

To ameliorate these shortfalls, our text generation process for human evaluation of summaries offers a span-based approach that resembles evidence annotation in question answering systems. A summary is thus a set of non-overlapping spans of contiguous text snippets from the chapter source. The total number of tokens across the spans is bound to the summary maximal token-length parameter. We considered twenty five readers from a book club as expert annotators, each assigned between ten to eleven distinct chapters for span labeling. We were less concerned about bias and avoided allocating more than one reader to a chapter.

In Table 12, we outline ROUGE F1 scores for human evaluation of span-based summaries. Top human scoring is at 0.72 ROUGE-L exceeding machine generation performance (Table 5) by up to about 2X. Given the current pace for developing state-of-the-art NLG systems, this apparent performance gap is expected to diminish rather precipitously, as research continues to reason the trade-off between cost and reward for conducting human annotation.

**Method Generalization** To evaluate the generalizability of our proposed two-step summarization method to other text genres or domains, we explored NarrativeQA (Kočiský et al., 2018). Destined for the reading comprehension (RC) problem space, NarrativeQA is a large-scale question answering dataset constructed from a collection of
large documents in the form of full-length books and movie scripts. Learning to understand books through effective summarization modeling become key to a successful RC system.

NarrativeQA comprises full-length books with an average of slightly over 60K tokens per story. While its human-curated abstractive summaries has a token complexity of about 650 on average. This suggests an end-to-end compression ratio of roughly 100 from source to summary. In contrast to our automatic method that yields a data compaction rate of close to 400 across the two computational steps on our biographical test set. We note that a NarrativeQA book is represented as a cohesive long sequence of text, rather than a collection of chapter entities like ours, the result of performing a data preprocessing step on each of our novels to improve model scalability.

The authors of NarrativeQA performed question answering quality experiments comparing the use of a book in its entirety to its labor-intensive human-created summary for retrieving an answer. Using the ROUGE-L metric they achieved 0.37 for summaries and 0.14 on full length stories. Although for a different goal, these results highly resemble our automatic evaluation scores of 0.38 and 0.12 for two-step and single-tier configurations, respectively.

7 Conclusion

In this paper, we presented a summarization approach that ensures hallucination-free text generation in its first step, and follows by a more regulated and manageable production of a final abstractive summary. On a biographical literary dataset with doubled to quadrupled chapter token complexity, our method achieved superior or similar performance compared to six baseline models. Empirical results show that our fact-unaware summarization can produce abstractive summaries with compelling factual consistency. Noting that author-created book descriptions are often of less than adequate quality, we encourage not only span-based but also free-form reader-written chapter summaries that are factually faithful and benefit a plausible load sharing for curating annotations.

Limitations

Our proposed summarization model is pretrained exclusively on news datasets, however, our experiments and analysis were conducted on biographical narratives. We only studied English summarization and our processes and in particular relevance findings are likely not entirely applicable to long multi-lingual documents. Moreover, single-domain trained models may propagate inductive biases rooted in the data they were pretrained on. This was evidenced in finetuning on our target dataset as the model demonstrated a moderate degree of transferability in adapting the newswire domain to our biographical discourse genre.

Our work studies generated summaries for long narrative text. While our taxonomy appears generalizable to other domains, investigating summarization quality of large-scale datasets, such as scientific articles, patent documents, government reports or meeting discourses was confined to the scope of baseline performance comparison.

Ethics Statement

We assembled our biographical dataset for the grounded source consistent with Project Gutenberg permissions and terms of use. Emanating personal identifiable information of the individual history is unavoidable when obtained from biographical literary. However, improving the faithfulness of automatically generated summaries is essential to ensure reliable and trusted factual accuracy. To the extent of our judgment, produced narrative summaries are free of harmful or offensive content, yet we plan to restrict our dataset for research use only.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful suggestions and feedback.

References


The distribution of discourse relations within and across turns in spontaneous conversation

S. Magalí López Cortez  Cassandra L. Jacobs
Department of Linguistics
University at Buffalo
Buffalo, NY, USA
solmagal;cxjacobs@buffalo.edu

Abstract

Time pressure and topic negotiation may impose constraints on how people leverage discourse relations (DRs) in spontaneous conversational contexts. In this work, we adapt a system of DRs for written language to spontaneous dialogue using crowdsourced annotations from novice annotators. We then test whether discourse relations are used differently across several types of multi-utterance contexts. We compare the patterns of DR annotation within and across speakers and within and across turns. Ultimately, we find that different discourse contexts produce distinct distributions of discourse relations, with single-turn annotations creating the most uncertainty for annotators. Additionally, we find that the discourse relation annotations are of sufficient quality to predict from embeddings of discourse units.

1 Introduction

Discourse relations (DRs) such as Elaboration, Background and Explanation, hold between discourse units contributing to the coherence of a text. Annotation of discourse relations has received attention for its relevance to discourse parsers, with applications in question answering systems (e.g. Jansen et al., 2014), text summarization (e.g. Liu and Chen, 2019), sentiment classification (e.g. Kraus and Feuerriegel, 2019), and machine translation (e.g. Meyer and Popescu-Belis, 2012). However, most of the annotated data and systems have focused on written language, with a few exceptions (e.g., Tonelli et al., 2010; Zeldes, 2017; Scholman et al., 2022). In spoken dialogue or multiparty conversation, participants must quickly juggle a variety of tasks, such as responding to another person to solve a problem (Levinson and Torreira, 2015) or negotiating the question under discussion (Roberts, 2012), often under considerable time pressure that is less present in written production. In addition to these time demands, it is unclear whether spontaneous conversation demonstrates the same patterns of discourse relations as observed in written language (see Crible and Cuenca, 2017, for a discussion of spoken vs. written use of discourse markers).

Perhaps unsurprisingly, the vast majority of work on discourse relations has focused either on written texts, especially news text (Carlson et al., 2003; Prasad et al., 2008, 2018), or highly structured conversations that are constrained by a particular game (Afantenos et al., 2015; Asher et al., 2016). Some recent corpora contain spoken monologues (Scholman et al., 2022), and spoken conversations (Tonelli et al., 2010; Zeldes, 2017), but the field still largely lacks annotated corpora of spontaneous dialogue.

Thus, our goal is to present the first efforts towards an annotated corpus of DRs for spontaneous spoken conversation, with particular attention to relations across different contexts within a conversation. We analyze the patterns of DR annotation within and across speakers and within and across turns and test the coherence of annotators’ decisions.

2 Related Work

Most currently available corpora annotated with DRs have focused on written language or spoken monologues. An exception is the George-town University Multilayer (GUM) corpus (Zeldes, 2017), which has a set of conversations annotated within Rhetorical Structure Theory (RST, Mann and Thompson, 1987), following the guidelines of the RST Discourse Treebank (RST-DT, Carlson et al., 2003). But it is an open question whether the DRs that have been identified for news texts are appropriate for conversational data. Tonelli et al. (2010) adopt the PDTB framework to annotate a subset of a corpus of Italian conversations about software and hardware troubleshooting, and suggest modifications to the framework to account for
Discourse relations corpora have usually been annotated by experts, but some recent corpora have been annotated by novice annotators, such as university students, in the case of the GUM corpus (Zeldes, 2017), or crowdsourced workers, in the case of the DiscoGEM corpus (Scholman et al., 2022). GUM was annotated using RST as part of a Corpus Linguistics class, while DiscoGEM was annotated following the Penn Discourse Treebank (PDTB, Prasad et al., 2008, 2018) framework, using a method for crowdsourcing annotations introduced in Yung et al. (2019), and using a multi-label approach. The present work deviates from prior work in its focus on conversational data and the use of Segmented Discourse Representation Theory (SDRT, Asher and Lascarides, 2003) alongside the STAC corpus (Asher et al., 2016) guidelines.

3 Discourse relation annotation

In this work, we focus on a subset of 19 dialogues from the Switchboard Corpus (Godfrey et al., 1992). This corpus contains informal language and has been the subject of study of numerous analyses of dialogue within linguistics (Jaeger and Snider, 2013; Reitter and Moore, 2014). In it, two strangers are presented with a topic (e.g., childcare) that they must discuss with each other, but the dialogues are otherwise not tightly constrained. Annotating Switchboard will provide us with a more complete understanding of the use and generality of discourse relations across linguistic contexts and genres.

Following the annotation procedure in the STAC corpus (Asher et al., 2016), we identified a subset of suitable elementary discourse units (EDUs) for annotation by parsing each turn into a dependency structure and included only those turns with at least two roots or verbs. Then, we segmented each of these turns into their respective EDUs. Using these segmentations, we identified EDU candidates for discourse relations that were either within-turn (same speaker) or across two turns (different speakers, or the same speaker), where the two turns were adjacent in the case of different speakers, or only interrupted by one turn, in the case of same speaker. We provide a representative set of these pair types in Table 1 under the Explanation, Comment, and Result examples, respectively.

3.1 Elementary Discourse Units

Elementary discourse units (EDUs) are typically defined as non-overlapping text spans (Mann and Thompson, 1987), which perform some basic discourse function (Asher and Lascarides, 2003), typically at the level of clauses. However, conversational EDUs may not necessarily contain a main verb (e.g., clarification questions: “Saginaw?”) or may be incomplete or interrupted (e.g., “and so—”). So, we define EDUs in Switchboard similarly to written text, with some modifications to account for variability due to spoken language. In particular, our modifications account for noise; non-linguistic communication (e.g., laughter); restarts; and disfluencies (e.g., “uh” or “um”). Additionally, we use complex discourse units (CDUs), which are combinations of EDUs which function together as an argument to a DR (Asher and Lascarides, 2003).

3.2 Relation categories

Discourse relations (DRs) were selected from Segmented Discourse Representation theory (SDRT, Asher and Lascarides, 2003), following the annotation manual for the STAC corpus (Asher et al., 2012). 11 out of 16 relation labels used in Asher et al. (2012) were selected, based on a pilot annotation. We selected the most common relations in an attempt to minimize the number of choices presented to annotators, but the set is non-exhaustive. An “Other” category was added for cases in which none of the selected labels applied. Table 1 shows the list of DRs together with representative examples.

3.3 Annotators

The present study recruited 114 students enrolled in a computational linguistics course grouped into 19 teams consisting of approximately 5 members who annotated the dyads. Each team received a conversation for annotation. Annotations were performed individually, but groups then discussed their work and submitted a report as a team. One team was excluded because they completed their annotations together and submitted a single set of labels. Students were trained to identify discourse relations using a short quiz and live training with the instructor of the course. Annotators were provided with guidelines to which they could refer back, and they had read and annotated the conversation in three previous tasks before annotating discourse relations, to ensure that they were familiar with the


<table>
<thead>
<tr>
<th>Relation</th>
<th>Discourse Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgement</td>
<td>A:</td>
</tr>
<tr>
<td></td>
<td>B:</td>
</tr>
<tr>
<td>Background</td>
<td>A:</td>
</tr>
<tr>
<td></td>
<td>B:</td>
</tr>
<tr>
<td>Clarification</td>
<td>A:</td>
</tr>
<tr>
<td>Question-Answer</td>
<td>B:</td>
</tr>
<tr>
<td></td>
<td>A:</td>
</tr>
<tr>
<td>Answer</td>
<td>B:</td>
</tr>
<tr>
<td>Explanation</td>
<td>A:</td>
</tr>
<tr>
<td></td>
<td>B:</td>
</tr>
<tr>
<td>Comment</td>
<td>B:</td>
</tr>
<tr>
<td></td>
<td>A:</td>
</tr>
<tr>
<td>Continuation</td>
<td>A:</td>
</tr>
<tr>
<td>Contrast</td>
<td>A:</td>
</tr>
<tr>
<td>Elaboration</td>
<td>A:</td>
</tr>
<tr>
<td></td>
<td>B:</td>
</tr>
<tr>
<td></td>
<td>A:</td>
</tr>
<tr>
<td>Result</td>
<td>B:</td>
</tr>
<tr>
<td></td>
<td>A:</td>
</tr>
<tr>
<td>Other</td>
<td>None of the labels applies</td>
</tr>
</tbody>
</table>

Table 1: Representative discourse unit pairs for annotated discourse relations. The first argument to the discourse relation is shown in italics and the second one in bold. A and B correspond to speakers, and double pipes (||) represent boundaries between elementary discourse units.

topics and speakers in each dyad.

3.4 Annotation procedure

Annotators were presented with pairs representing either an EDU or CDU ($\pi_1$) and another EDU or CDU ($\pi_2$). Annotators were shown two spans of text $\pi_1$ and $\pi_2$ with $\pi_1$ presented in italics and $\pi_2$ presented in bold face font in the annotation software Prodigy (Montani and Honnibal, 2018), with two preceding and two subsequent turns for context. Annotators were asked to determine the relation between $\pi_1$ and $\pi_2$ from a list of the DR categories in Table 1. If annotators thought that no relation was present, they were told to reject the item and move on to the next pair. Critically for our research question, annotators could mark several relations for a pair of EDUs simultaneously. In addition to labeling discourse relations, annotators were also asked to provide a confidence rating on a scale from 1-5, but we leave these analyses for future work. In total, each annotator provided judgments for an average of 25 EDU pairs across 464 total pairs.

In the next section, we test whether annotators show greater uncertainty about discourse relations in different discourse contexts. We analyze the distribution of their labels to assess whether discourse relations in conversation vary in their contexts of use.

4 Uncertainty in the annotation of discourse relations

Different EDU pairs in the present annotation task were drawn either from the same turn, or across turns but within or across speakers. Thus, we can assess how much discourse relations vary by the placement of an utterance in a dialogue. Given the complex dynamics in dialogue, we expect to find significant differences in discourse relation use across different discourse contexts. We visualize the distribution of the relations in Figure 1.

Annotators generally selected more discourse relations per EDU pair in the single-turn case, with an average of 8.16 relations per team or 1.60 per annotator. When EDUs spanned turns within a single speaker, groups selected significantly fewer relations (average = 7.29, $t$(302) = −2.16, $p < .05$). Groups likewise selected even fewer relations for EDUs between two speakers (average = 6.51, $t$(314) = −2.54, $p < .05$). On its face, this pattern appears surprising, because it suggests that annotators find more relations appropriate for single-speaker productions. However, an alternative interpretation of these results is that annotators may instead have been uncertain about the distinctions between the different discourse relations. This second interpretation is corroborated by post-hoc poll data from 35 annotators, of whom 32 (91.4%) stated that the selection of discourse relations was best suited to annotating cross-speaker...
EDU pairs. Future work will require recruiting greater numbers of annotators to be able to distinguish between these two hypotheses.

4.1 Inter-annotator agreement

We computed measures of inter-annotator agreement for multilabel tasks using Marchal et al. (2022). This approach uses bootstrap sampling to estimate the chance frequencies of DRs in a multilabel dataset to provide a baseline for agreement between annotators.

We summarize the results of this analysis in Table 2. Following Marchal et al. (2022), we computed observed, expected and adjusted agreement for six measures. Soft-match agreement uses the intersection of labels selected by two annotators; boot-match corrects the expected agreement by using the bootstrapping method (as opposed to ignoring non-intersecting labels); augmented kappa uses DR labels weighted according to the number of labels annotated for each item; precision and recall are calculated as the proportion of intersecting DR labels over the set of labels selected by the first and second annotator, respectively; F1 is the usual harmonic mean between precision and recall.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Intercept</th>
<th>Different speaker</th>
<th>Within turn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>-2.73</td>
<td>1.96</td>
<td>0.24</td>
</tr>
<tr>
<td>Clarification Q.</td>
<td>0.02</td>
<td>0.41</td>
<td>-0.43</td>
</tr>
<tr>
<td>Comment</td>
<td>-1.69</td>
<td>0.84</td>
<td>0.46</td>
</tr>
<tr>
<td>Continuation</td>
<td>-1.60</td>
<td>2.35</td>
<td>0.31</td>
</tr>
<tr>
<td>Contrast</td>
<td>-2.23</td>
<td>2.07</td>
<td>0.17</td>
</tr>
<tr>
<td>Elaboration</td>
<td>-0.78</td>
<td>1.95</td>
<td>0.13</td>
</tr>
<tr>
<td>Explanation</td>
<td>-1.40</td>
<td>2.00</td>
<td>0.09</td>
</tr>
<tr>
<td>Narration</td>
<td>-3.29</td>
<td>2.68</td>
<td>0.53</td>
</tr>
<tr>
<td>Other</td>
<td>-3.54</td>
<td>2.20</td>
<td>1.12</td>
</tr>
<tr>
<td>Q-A Pair</td>
<td>-0.81</td>
<td>0.64</td>
<td>-0.01</td>
</tr>
<tr>
<td>Result</td>
<td>-2.35</td>
<td>1.63</td>
<td>-0.00</td>
</tr>
</tbody>
</table>

Table 3: Coefficient estimates from a multiclass logistic regression predicting each annotation label.

2010), and partly due to annotators’ uncertainty on DR labels across different context types.

4.2 Predicting relation selection

We use a model comparison approach to understand the contributions of discourse context (within/across speakers and within/across turns) to relation annotation by first constructing a null model that estimates the base rates of each dis-
course relation. Then, we constructed a multiclass logistic regression model containing the discourse context variables of interest, which significantly improved fit to the annotation data ($X^2(22) = 447.98, p < .001$). This improvement in fit suggests that the distribution of discourse relations that are identified by annotators is distinct across contexts. Adding the annotator group/topic also significantly improved fit beyond the model containing the contextual variables alone ($X^2(198) = 900.06, p < .001$). We summarize the results of this final model in Table 3.

An informal evaluation of the coefficients suggests that discourse relations are not uniformly distributed across contexts. Intuitively, Acknowledgements, Clarification Questions, Comments, and Question-Answer Pairs are more likely across speakers than within. Additionally, Continuations, Elaborations, Explanations, and Narrations are more likely to occur within a single speaker. The pattern of results is more unclear when comparing EDUs that are produced by a single speaker but which occur either within or across turns. For example, relations such as Clarification Questions are less likely to occur within a turn than across turns.

### 4.3 Classifier for relations

To validate the quality of the annotations, we built a model to classify EDU pairs into discourse relations. We reasoned that if annotators are following the guidelines and use information about the EDU pairs, then a classifier should be able to predict DR labels. We encoded the first EDU or CDU ($\pi_1$) and the second ($\pi_2$) as the two “sentences” in the next sentence prediction architecture of BERT (Devlin et al., 2019). This enables the classifier to represent the $\pi_1$ and $\pi_2$ components somewhat separately.

We built a classifier head trained on the resulting embeddings without fine-tuning to predict each individual annotator label. We chose to model each annotator label individually to learn agreement/majority class implicitly because prior studies have shown that this improves generalization (Yung et al., 2022). We use a leave-one-conversation-out training procedure, in which we test a ridge regression classifier on all of the annotations from a single conversation while we train it on all other annotations across the other conversations. This ensures minimal memorization of specific turns within a conversation, which is critical given our multilabel annotation approach.

Strict annotation-level accuracy to predict each selected label from all annotators was quite poor, with macro average precision at .21, recall at .19, and F1 at .19. However, recall was substantially higher when considering whether the top guess belonged to the set of all labels provided by annotators, at .76 overall and .71 averaged by group.

To quantify the uncertainty of the annotators across different contexts, we leverage the classifier to produce a label distribution for a given ($\pi_1, \pi_2$) pair. We then compute the cross-entropy between the model’s predictions and annotators’ gold label distributions, collapsing across all annotations for an EDU pair. Overall, cross entropy between model predictions and annotator labels was highest for the single-turn case, with (mean $= 0.43$), but lowest for EDUs between two speakers (mean $= 0.38$), suggesting greater uncertainty in label assignment.

### 5 Discussion

In two experiments, we demonstrated that novice DR label annotations in a single turn are more difficult than across turns. We showed that including discourse context (within/across speaker and within/across turn) to a logistic regression model significantly improves fit to our annotation data. A classifier trained to predict DR labels from embeddings of ($\pi_1, \pi_2$) pairs showed modest success for recall of any of the annotations, but poor precision and recall overall. A comparison of this classifier’s predictions and annotators’ gold label distributions revealed greater uncertainty for the annotation of discourse relations within a single turn.

These results demonstrate that different conversational contexts are associated with different distributions of discourse relations. The uncertainty of choice of discourse relations within a turn may be due to several factors. DRs that typically occur across adjacent turns and across speakers (e.g., Acknowledgements) might have clearer signals. At the same time, DRs that occur more frequently within speakers, and, in particular, within a turn, might be more ambiguous, or might co-occur with other relations. More work is necessary to disentangle uncertainty about the identity of the best fit relation from whether multiple relations are appropriate.
Limitations

The current work is limited by the size of the dataset and the nature of spontaneous conversation. While the discourse relations proposed as part of this work were selected to be general and build on categories from the literature, the list is not exhaustive and it is likely that these relations may be culturally, linguistically, and situationally specific. Future work in this area should validate the generality of the discourse relation system used in this work.

The selection of EDUs and CDUs for annotation is also non-exhaustive; additional segments could be included in future work.

Annotation quality is also a practical limitation. Annotation for discourse relations typically results in low-agreement data, even among expert annotators (e.g., DiscoGEM; Scholman et al., 2022). Even though our research questions focus on this disagreement as a positive, other researchers may require greater numbers of annotations in order to obtain a gold label.

Ethics Statement

We are not aware of ethical issues associated with the texts used in this work. Students participated in the annotation task as part of course credit but annotation decisions were not associated with their performance in the course.

Acknowledgements

We would like to thank Jürgen Bohnemeyer and three anonymous reviewers for feedback on a previous version of this paper.

References


Embedding Mental Health Discourse for Community Recommendation

Hy Dang, Bang Nguyen, Noah Ziems, Meng Jiang
University of Notre Dame
{hdang, bnguyen5, nziems2, mjiang2}@nd.edu

Abstract

Our paper investigates the use of discourse embedding techniques to develop a community recommendation system that focuses on mental health support groups on social media. Social media platforms provide a means for users to anonymously connect with communities that cater to their specific interests. However, with the vast number of online communities available, users may face difficulties in identifying relevant groups to address their mental health concerns. To address this challenge, we explore the integration of discourse information from various subreddit communities using embedding techniques to develop an effective recommendation system. Our approach involves the use of content-based and collaborative filtering techniques to enhance the performance of the recommendation system. Our findings indicate that the proposed approach outperforms the use of each technique separately and provides interpretability in the recommendation process.

1 Introduction

The rise of social media as a platform has allowed people all over the world to connect and communicate with one another. Further, these communities that exist online are able to keep their members anonymous from one another, allowing new communities to form which would have a hard time existing without anonymity.

Specifically, this new and robust anonymity has allowed an explosion of online communities with a focus on giving each other advice on health issues. While being involved in seeking peer support in a community with people that have experienced similar issues can provide a significant positive impact on someone’s ability to navigate their personal problems (Richard et al., 2022), finding communities with relevant discourse is not trivial. Often, the platforms which host these communities have a very large quantity of them. There are over 100,000 different communities on Reddit alone. Further, some communities are not easily found due to their inherently anonymous nature, so the only way a user can decide if they fit within the community is by spending time reading through the discourse happening within the community.

For these reasons, new users seeking others who have experienced similar situations may have a very hard time finding communities that would help them the most, even if they are familiar with the platform which hosts the communities.

Recently, embedding long sequences of text has received lots of interest both from the research community and from practitioners. A number of studies have shown embeddings can be useful for measuring the similarity both between document pairs and between question-document pairs (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2021), allowing for retrieval of the most similar documents given a new question or document. However, little work has been done investigating how the discourse within a community, which represents the meaning of that community, can be represented in a single embedding. The discourse of a community in this context can be all users’ posts in that specific community or represented community’s description. This poses a unique challenge as discourse within these communities is often in the form of threads that, unlike documents, are not naturally represented as a single block of text.

The goal of this work is to develop a system to recommend support groups to social media users who seek help regarding mental health issues using embeddings to represent the communities and their discourse. Specifically, we aim to leverage the text of a given user’s posts along with the description and posts in each subreddit community to help recommend support groups that the user could consider joining.

Our main research questions are as follows:
1. In representing online communities through discourse embeddings, what type of information can be used?

2. To what degree do these representations improve the accuracy of predicting users’ behaviors regarding their involvement in sharing experiences within groups or communities?

3. Do different discourse embedding methods change the prediction capacity of our community recommendation model?

In exploring these research questions, we propose a hybrid recommendation approach that leverages both content-based and collaborative filtering to construct our community recommendation model. As shown in Fig. 1, the content-based filtering component investigates different methods of embedding discourse within a community to recommend similar communities to users. It is then combined with a matrix factorization model that learns user engagement behavior in a community to improve recommendation decisions. Utilizing users’ past interactions as well as text-based information about the communities, we show that our model achieves promising accuracy while offering interpretability.

2 Related Work

There are a number of studies related to our work. Son et al. (2022) and Balusu et al. (2022) constructed discourse embeddings to find relations between short text segments. While the two studies were similar in concept, they focused on short text segments where this work instead focused on constructing discourse embeddings for entire social media communities.

Garriga et al. (2022) showed NLP techniques could be used with electronic health records to predict mental health crises 4 weeks in advance. While online communities were no replacement for professional medical help, this suggested many who had looming mental health problems seek help before a crisis.

Low et al. (2020) experimented on the same dataset we used with Natural Language Processing techniques such as TF-IDF and sentiment analysis to understand the effects of COVID-19 on mental health. Although working on the same dataset, our work studies a different task: to recommend mental health-related support community to Reddit users.

Musto et al. (2016) adopted a similar approach to ours in content-based filtering for recommendation. Specifically, they mapped a Wikipedia page to each item and generate its corresponding vector representation using three feature-extraction methods - Latent Semantic Indexing, Random Indexing, and Word2Vec. We extended this method by exploring more recent representations of text such as BERT (Devlin et al., 2019) and OpenAI embeddings.

Halder et al. (2017) recommended threads in health forums based on the topics of interest of the users. Specifically, self-reported medical conditions and symptoms of treatments were used as additional information to help improve thread recommendations (Wang et al., 2020; Jiang et al., 2012). While our work is also situated in the health domain, we are interested in recommending a broader support group to users rather than a specific thread.

Ghazarian et al. (2022) used sentiment and other features to automatically evaluate dialog, showing NLP techniques could be used to evaluate quality of discourse. In doing so, they leveraged weak supervision to train a model on a large dataset without needing quality annotations.

3 Problem Definition

Suppose we have a Reddit’s "who-posts-to-what" graph, which is denoted by \( G = (U, V, E) \) where \( U \) is the set of users, \( V \) is the set of subreddit communities, and \( E \), a subset of \( U \times V \), is the set of edges. The number of user nodes is \( m = |U| \) and the number of subreddit communities is \( n = |V| \). So, \( U = \{u_1, u_2, P_1, P_2, ..., u_m, P_m\} \) where \( P_i \) is the set of posts by user \( u_i \) and \( V = \{v_1, P'_1, v_2, P'_2, ..., v_n, P'_n\} \) where \( P'_j \) is the set of all posts in subreddit \( v_j \).

If a user \( u_i \) posts to subreddit \( v_j \), there is an edge that goes from \( u_i \) to \( v_j \), which is denoted by \( e_{ij} = e(u_i, v_j) \). The problem is that given \( G \), predict if \( e_{ij} = e(u_i, v_j) \) exists. In other words, will user \( u_i \) post to subreddit \( v_j \)?

4 Methodology

Figure 1 illustrates our recommendation pipeline, which adopts a hybrid approach by incorporating both content-based filtering (CBF) and collaborative filtering, specifically matrix factorization (MF) strategies. The CBF model recommends new subreddits based on the average of a user’s previous interactions, weighted by how similar the previous subreddits are to the new ones. Meanwhile, users and subreddits are represented in a \( k \)-dimensional...
joint latent space in the MF model. The distance between users and subreddits in this latent space is used to provide recommendations for new subreddits. The predictions from these two components are linearly combined to obtain the final recommendation of subreddits to users.

The collaborative filtering component of our solution leverages nonnegative matrix factorization to represent our users and subreddits in lower-dimensional latent space. In this sense, we redefine the adjacency matrix $A$ in our problem definition so that it works with nonnegative factorization. More specifically, users’ past interactions with items are represented by the adjacency matrix $A \in \{5, 1, 0\}^{m \times n}$. $A_{ij} = 5$ if the user $u_i$ has posted to subreddit $j$, $A_{ij} = 1$ if the user $u_i$ has NOT posted to the subreddit $v_j$, and $A_{ij} = 0$ is the missing connection that needs predicting. Given this adjacency matrix $A$, the task is to predict the missing elements $A_{ij} = 0$. In the following sections, we elaborate on each component of our recommendation model and then discuss how they are combined to obtain our final solution.

### 4.1 Content-based Filtering

In recommending items to users based on their past interactions and preferences, content-based filtering methods represent each item with a feature vector, which can then be utilized to measure the similarity between items (Linden et al., 2003). If an item is similar to another item with which a user interacted in the past, it will be recommended to that same user. Thus, in addition to the adjacency matrix $A$, we utilize another matrix $C$ of size $m \times m$, where $C_{ab}$ is the similarity between the embeddings for two subreddits with embedding vectors $a$ and $b$. In this paper, we use cosine similarity as the similarity measure:

$$C_{ab} = \frac{a \cdot b}{\|a\| \cdot \|b\|}.$$ 

To predict the value of the missing element where $A_{ij} = 0$ (whether user $u_i$ will post to subreddit $v_j$), we compute the average of user $u_i$’s past interactions (which subreddits user $u_i$ posted and did not post to), weighted by the similarity of these subreddits to subreddit $v_j$. Mathematically,

$$A'_{ij} = \frac{\sum_{k=1}^{n} A_{ik} C_{kj}}{\sum_{k=1}^{n} C_{kj}}.$$ 

We can generalize the above formula to obtain the new predicted adjacency matrix using matrix-level operations:

$$A^{(CBF)} = (AC) \odot D,$$

where...
\[ \mathbf{D} = 1 / (\mathbf{I} \cdot \mathbf{C}) \text{ (element-wise)}, \]
\[ \mathbf{I} \text{ is an indicator matrix such that } I_{ij} = 1 \text{ if } A_{ij} \neq 0, \text{ otherwise } I_{ij} = 0, \]
\[ \text{and } \odot \text{ is the Hadamard product.} \]

**4.1.1 Representing Subreddit Discourse with Description and Posts**

It is helpful to consider the specific domain of the application to represent each item as an embedding. In the context of our subreddit recommendation problem, we take advantage of two types of text-based information about a subreddit to construct the similarity matrix: (1) the posts within the subreddit itself and (2) the general description about the subreddit provided by the subreddit moderators.

We then use a feature extraction method to obtain two embeddings of a subreddit, one based on its description and the other based on its posts. As a subreddit contains many posts, each of which has a different embedding given the same feature-extraction method, we take the average of the embeddings across all posts within a subreddit to obtain one embedding for the subreddit.

**4.1.2 Feature Extraction**

In this paper, we consider three feature-extraction methods: Term Frequency-Inverse Document Frequency (TF-IDF), Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), and OpenAI.*

**TF-IDF**: The TF-IDF algorithm represents a document as a vector, each element of which corresponds to the TF-IDF score of a word in that document. The TF-IDF score for each word in the document is dictated by (1) the frequency of the word in the document (Sparck Jones, 1972), and (2) the rarity of the word in the entire text corpus (Luhn, 1957). That is, a term is important to a document if it occurs frequently in the document but rarely in the corpus. We use the implementation from scikit-learn (Pedregosa et al., 2011) to obtain the TF-IDF representations of our subreddits.

**BERT**: We employ BERT to generate sentence embeddings as another feature extraction technique (Devlin et al., 2019). BERT takes a sentence as input and generates a fixed-length vector representation of the sentence. This representation is meant to capture the syntactic and semantic meaning of the input sentence in a way that can be used for various natural language processing tasks, such as sentence classification or semantic similarity comparison. In the context of our problem, we can treat each subreddit description or each post as a sentence and feed it to a pre-trained BERT model to generate the embeddings that represent the subreddit. Long posts are truncated to fit within the context limits of pre-trained models. We experiment with 4 different variations of BERT embeddings:

- BERT base and large (Devlin et al., 2019)
- Sentence-BERT, or SBERT (Reimers and Gurevych, 2019)
- BERTweet (Nguyen et al., 2020)

**OpenAI**: Similar to BERT embeddings, OpenAI embeddings take in a string of text and output an embedding that represents the semantic meaning of the text as a dense vector. To do this, the input string is first converted into a sequence of tokens. The tokens are then fed to a Large Language Model (LLM), which generates a single embedding vector of fixed size. OpenAI’s text-embedding-ada-002 can take strings of up to 8191 tokens and returns a vector with 1536 dimensions.

**4.2 Nonnegative Matrix Factorization for Collaborative Filtering**

Matrix factorization (MF) approaches map users and items (subreddits in this case) to a joint latent factor space of a lower dimension \( k \) (Koren et al., 2009). The goal of this method is to recommend to a user the subreddits that are close to them in the latent space. More formally, MF involves the construction of user matrix \( \mathbf{P} \) of dimension \( n \times k \) and subreddit matrix \( \mathbf{Q} \) of dimension \( m \times k \). In this sense, the resulting term, \( \mathbf{p}_i^\top \mathbf{q}_j \), captures user \( u_i \)'s interest in item \( v_j \)'s characteristics, thereby approximating user \( u_i \)'s rating of item \( v_j \), or denoted by \( A_{ij} \).

This modeling approach learns the values in \( \mathbf{P} \) and \( \mathbf{Q} \) through the optimization of the loss function

\[
\min_{\mathbf{P}, \mathbf{Q}} \sum_{A_{ij} \in \mathbf{A}} (A_{ij} - \mathbf{p}_i^\top \mathbf{q}_j)^2 + \lambda (\|\mathbf{p}_i\|^2 + \|\mathbf{q}_j\|^2).
\]

Matrix factorization offers the flexibility of accounting for various data and domain-specific biases that may have an effect on the interaction between user \( u_i \) and subreddit \( v_j \). In this paper, we consider three types of biases: global average \( \mu \),

---

*OpenAI API Embeddings: https://platform.openai.com/docs/guides/embeddings
user bias \(b_i^{(p)}\), and subreddit bias \(b_j^{(q)}\). The updated loss function is given by:
\[
\min_{P,Q} \sum_{A_{ij} \in A} (A_{ij} - \mu - b_i^{(p)} - b_j^{(q)})^2 + \\
\lambda(||p_i||^2 + ||q_j||^2 + b_i^{(p)} + b_j^{(q)})^2).
\]
(1)

After optimization, each element in the new predicted adjacency matrix \(A^{\text{MF}}\) is given by:
\[
A_{ij}^{(\text{MF})} = p_i^T q_j + \mu + b_i + b_j
\]

### 4.3 Final Model: Hybrid Approach

Our main model leverages insights from both content-based filtering and matrix factorization by taking a linear combination of their predicted adjacency matrix. Specifically, the new adjacency matrix is given by:
\[
A^{(\text{MF+CBF})} = \beta A^{(\text{CBF})} + (1 - \beta) A^{(\text{MF})},
\]
where \(\beta\) is a hyperparameter that controls how much the CBF model (vs MF model) contributes to the final prediction.

### 5 Data and Experimental Setup

For the experimental setup, we use the data from Low et al. (2020) working on Reddit platforms in mental health domains, particularly health anxiety.

#### 5.1 Data Description

The dataset is collected from 28 mental health and non-mental health subreddits. The dataset is suitable for studying how subreddits and social media platforms correlated with individuals’ mental health and behavior. The original data comprises 952,110 Reddit posts from 770,176 unique users across 28 subreddit communities, which include 15 mental health support groups, 2 broad mental health subreddits, and 11 non-mental health subreddits. We also manually collect descriptions of the 28 subreddits and use that information along with the posts to conduct the content similarity matrix.

#### 5.2 Data Preprocessing

Although the original dataset has a large number of unique users, the majority of them only contribute posts to one or two different communities. This presents a challenge when evaluating our specific task. As our objective is to examine users’ behavior over time and provide recommendations for engaging in suitable subreddits, we have implemented a filter to exclude users who post to fewer than three subreddits. After filtering, the remaining users and posts are 16,801 and 69,004, respectively, while the number of subreddits remains to be 28.

We also seek to understand the distribution of interactions between users and different subreddits. The detailed distribution of post frequency across subreddits is visualized in Figure 2.

![Figure 2: Distribution of post frequency across subreddits](image-url)

#### 5.3 Experimental Setup

##### 5.3.1 Data Splits

To construct our data splits, for each user in our dataset, we choose the most recent subreddit that the user first posted to as the test example. For example, if the user post history is [subreddit1, subreddit2, subreddit3, subreddit1, subreddit2], then subreddit3 will be used as the test example. For each positive training example, we pair it with a negative example randomly sampled from the list of subreddits where the user has not posted.

##### 5.3.2 Evaluation Metrics

In assessing the performance of our recommendation method and the baseline, we use the following evaluation metrics: Recall@K and Mean Reciprocal Rank (MRR).

#### 5.4 Results

Table 1 presents the performance of our hybrid recommendation system as well as its individual relationships.
components (MF or CBF). For CBF, we report its performance on different types of embeddings constructed using different information (posts or description) and different feature extraction methods (TF-IDF, BERT, or OpenAI). Figure 3 visualizes the results of exemplary models in a diagram for better analysis using Recall@K.

According to Table 1, all variants of our recommendation method outperform the random predictor. Among all the variants, the hybrid solution using the content similarity matrix generated from OpenAI embeddings achieves the highest performance in MRR (0.4244) and average Recall@K.

For CBF, operating a feature-extraction method on subreddit posts results in higher performance than operating the same method on description. For example, the MRR for CBF - BERT base is 0.3140 when using posts and 0.3024 when using description. It can also be observed that given the same information (either posts or information), deep-learning-based feature extraction methods like OpenAI and BERT bring about better performance for CBF than TF-IDF.

As our recommendation model combines both MF and CBF, we investigate the effect of hyper-parameter $\beta$, which dictates how much CBF contributes to the final prediction. Figure 4 illustrates the performance of the hybrid models on varying $\beta$. When $\beta = 0$, the hybrid model’s performance is the same as that of MF. When $\beta = 1$, the hybrid model’s performance is the same as that of CBF. It can be seen from the peak of these curves that this way of linearly combining MF and CBF brings about significant improvement in MRR.

5.5 Case Studies

We perform a series of case studies to understand why certain information and methods are more helpful than others in recommending subreddits to users. We present our findings by comparing the behavior of the following models: (1) CBF models using TF-IDF and OpenAI Embedding on Subreddit Descriptions, (2) CBF models using OpenAI Embeddings on Subreddit Descriptions and Posts, and (3) MF model and Hybrid model.

5.5.1 CBF models using TF-IDF and OpenAI Embedding on Subreddit Descriptions

The objective of the first case study is to investigate the impact of different types of embedding methods on the performance of recommendations. To achieve this, we employ TF-IDF and OpenAI Embedding approaches to analyze subreddit descriptions and compare their predictions using content-based filtering (CBF) approaches, as illustrated in Figure 5. Specifically, we consider User A’s historically interacted subreddits, which relate to depression, loneliness, and anxiety, respectively, with the ground truth of social anxiety. For CBF models, the content similarity $C$ between historically interacted and ground truth subreddits is crucial for accurate predictions. Hence, we evaluate the similarity scores between them. According to the result, the OpenAI Embedding technique outperforms TF-IDF in learning the representation of subreddits. Based on the analysis of content similarity matrices of the two approaches, we observe that TF-IDF has low similarity scores among subreddits due to its bag-of-words (BOW) approach, which fails to capture semantic relationships in short texts (Naseem et al., 2021), such as subreddit descrip-
To achieve this goal, we evaluate OpenAI Embeddings on posts and descriptions. Figure 6 illustrates the predictions using CBF approaches utilizing OpenAI Embeddings on posts and descriptions. Specifically, we examine User B’s historical posts, which are in depression and personal finance, and the ground truth label is legal advice. To understand the behavior of CBF on these two types of information, we analyze the similarities between historical subreddit interactions of User B and how the ground truth label is correlated with these subreddits. Our analysis shows that using OpenAI Embeddings on subreddit posts can capture strong relationships between personal finance and legal advice, where many legal advice posts are related to financial information. However, when only using subreddit descriptions of legal advice, which is "A place to ask simple legal questions, and to have legal concepts explained.", the model fails to capture this relationship. Furthermore, as shown in Table 1, the use of subreddit posts as representations for communities generally exhibits higher performance across most metrics when compared to using community descriptions. The reason is that subreddit descriptions contain less information than posts describing only the general purpose of the subreddit. In contrast, using subreddit posts can accurately learn the representations of the subreddits. Therefore, among the two types of information, using subreddit posts to represent subreddits helps models achieve better performance.

<table>
<thead>
<tr>
<th>Approach</th>
<th>MRR</th>
<th>Recall@1</th>
<th>Recall@3</th>
<th>Recall@5</th>
<th>Recall@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Predictor</td>
<td>0.1631</td>
<td>0.0429</td>
<td>0.1318</td>
<td>0.2221</td>
<td>0.4409</td>
</tr>
<tr>
<td>Matrix Factorization (MF)</td>
<td>0.3895</td>
<td>0.2300</td>
<td>0.4197</td>
<td>0.5585</td>
<td>0.7946</td>
</tr>
<tr>
<td>CBF - TF-IDF (Description)</td>
<td>0.2751</td>
<td>0.1503</td>
<td>0.2777</td>
<td>0.3634</td>
<td>0.5494</td>
</tr>
<tr>
<td>CBF - BERT base (Description)</td>
<td>0.3024</td>
<td>0.1807</td>
<td>0.3050</td>
<td>0.3799</td>
<td>0.5668</td>
</tr>
<tr>
<td>CBF - OpenAI (Description)</td>
<td>0.3113</td>
<td>0.1761</td>
<td>0.3233</td>
<td>0.4266</td>
<td>0.6093</td>
</tr>
<tr>
<td>CBF - SBERT (Post)</td>
<td>0.2865</td>
<td>0.1317</td>
<td>0.3109</td>
<td>0.4281</td>
<td>0.6545</td>
</tr>
<tr>
<td>CBF - BERT base (Post)</td>
<td>0.3140</td>
<td>0.1598</td>
<td>0.3446</td>
<td>0.4776</td>
<td>0.6651</td>
</tr>
<tr>
<td>CBF - BERT large (Post)</td>
<td>0.3168</td>
<td>0.1637</td>
<td>0.3436</td>
<td>0.4795</td>
<td>0.6674</td>
</tr>
<tr>
<td>CBF - BERTweet base (Post)</td>
<td>0.3154</td>
<td>0.1570</td>
<td>0.3516</td>
<td>0.4918</td>
<td>0.6700</td>
</tr>
<tr>
<td>CBF - OpenAI (Post)</td>
<td>0.3195</td>
<td>0.1642</td>
<td>0.3484</td>
<td>0.4815</td>
<td>0.6823</td>
</tr>
<tr>
<td>MF + CBF OpenAI (Description)</td>
<td>0.4039</td>
<td>0.2405</td>
<td>0.4491</td>
<td>0.5790</td>
<td>0.8093</td>
</tr>
<tr>
<td>MF + CBF BERT base (Post)</td>
<td>0.4114</td>
<td>0.2449</td>
<td>0.4613</td>
<td>0.5966</td>
<td>0.8023</td>
</tr>
<tr>
<td>MF + CBF BERTweet base (Post)</td>
<td>0.4221</td>
<td>0.2570</td>
<td>0.4809</td>
<td>0.6022</td>
<td>0.8056</td>
</tr>
<tr>
<td>MF + CBF BERT large (Post)</td>
<td>0.4237</td>
<td><strong>0.2593</strong></td>
<td>0.4832</td>
<td>0.6000</td>
<td>0.8059</td>
</tr>
<tr>
<td>MF + CBF OpenAI (Post)</td>
<td><strong>0.4244</strong></td>
<td>0.2571</td>
<td><strong>0.4841</strong></td>
<td><strong>0.6063</strong></td>
<td><strong>0.8154</strong></td>
</tr>
</tbody>
</table>

Table 1: Model Performance with different content similarity matrices generated by embedding methods evaluated on MRR and Recall@K
5.5.3 MF vs MF + CBF model using OpenAI Embeddings on Subreddit Discourses

The objective of the third study is to investigate the performance improvement achieved by combining MF and CBF. Specifically, we aim to explore how the use of discourse embeddings can generate content similarity matrices among subreddits that can address challenges encountered by the MF approach. To this end, we evaluate the MF and MF + CBF approaches using OpenAI Embeddings on posts. The predictions generated by the two models are presented in Figure 7.

We further examine the construction of scores using MF for this case study. The scores values are generated using latent features $P$, $Q$, $u$, $b(p)$, and $b(q)$, representing user, item features, global average, user, and item biases, respectively. However, due to the imbalance in the dataset, there are more posts in some subreddits than others, leading to a cold start problem for the MF approach to accurately learn communities with a small number of examples. In this case study, MF fails to generate correct predictions for the divorce community due to the limited number of posts available. Additionally, MF is biased towards subreddits with more posts, as reflected by the $b(q)$ values that have strong correlations with the number of posts in the subreddit communities, as depicted in Figure 8.

We demonstrate that the top three predictions generated by MF are the subreddits with the highest item biases compared to other subreddits, which are also the ones with the most posts. However, as divorce only accounts for 0.78% of the dataset, the performance of MF is limited. By utilizing OpenAI Embeddings on Subreddit Discourses to represent subreddit communities, we can integrate semantic information into the prediction process, thereby overcoming the cold start problem encountered by MF. Furthermore, this approach captures the relationships between the target recommended subreddit, historically interacted communities and semantic similarities. In this case, the most similar subreddits to personalfinance are legaladvice and divorce, while the most similar subreddits to parenting are autism and divorce.

Overall, we showcase that integrating semantic information into MF can address the cold start problem, and combining MF with CBF using discourse embeddings can make better recommendations.
6 Conclusion

This study aimed to investigate the effectiveness of different types of discourse embeddings when integrated into content-based filtering for recommending support groups, particularly in the mental health domain. Our findings showed that the hybrid model, which combined content-based filtering and collaborative filtering, yielded the best results. Moreover, we conducted an extensive case study to demonstrate the interpretability of our approach’s predictions.

Previous studies have brought to light the use of past behaviors to make more accurate recommendations in mental health (Valentine et al., 2022). They also emphasize effective communication between the recommender system and the user as an essential factor for users’ proper understanding of mental health in general as well as in their own journey (Valentine et al., 2022). Through promising prediction accuracy and interpretability, we believe that this method can serve as a valuable tool to support individuals, particularly those with mental health concerns, to share and seek help regarding their issues.

Limitations

In our current project, we have not taken into account the temporal information that treats the historical behavior of users as a sequence of actions. Thus, the model may not capture how user behaviors change over time. To ensure full support to users in need, we recommend that future work should address this limitation by considering users’ historical behaviors as a sequence of actions. Moreover, although our pre-trained models achieved significant results without fine-tuning discourse embeddings, we suggest that fine-tuning these models can enhance performance by capturing the nuances of the datasets’ distribution and contexts. Furthermore, conducting a detailed comparison of additional open-source Large Language Models (LLMs) would provide more comprehensive insights into their performance. Additionally, in addition to analyzing the efficiency of different models, it is crucial to evaluate the cost associated with implementing these models. Therefore, future work should consider both fine-tuning and evaluating additional LLMs, while also taking into account the costs of utilizing these models.

Acknowledgement

This work was supported by NSF IIS-2119531, IIS-2137396, IIS-2142827, CCF-1901059, and ONR N00014-22-1-2507.

References

Murali Raghu Babu Balusu, Yangfeng Ji, and Jacob Eisenstein. 2022. Pre-trained sentence embeddings for implicit discourse relation classification.


ACM international conference on Information and knowledge management, pages 45–54.


Abstract
We present a corpus of parallel German-language simplified newspaper articles. The articles have been aligned at sentence level and annotated according to the Rhetorical Structure Theory (RST) framework. These RST annotated texts could shed light on structural aspects of text complexity and how simplifications work on a text level.

1 Introduction
The goal of text simplification is to reduce the complexity of a text whilst retaining the main information, in order to make a text easier to understand. In this paper, we present a corpus of German-language parallel simplified newspaper articles, at three different complexity levels. Each text has been annotated according to the Rhetorical Structure Theory (RST, Mann and Thompson, 1988) framework. RST posits that segments (or Elementary Discourse Units, EDUs) in a text are related to each other and that one component of the pair serves as a ‘nucleus’ and the other as a ‘satellite’, or in some cases both components are considered to have ‘nucleus’ status. An example of a relation that connects EDUs is ‘evidence’ where the nucleus is a claim and the satellite provides evidence for the claim. RST therefore provides information about the structure of texts; an area which has thus far not been the focus of much research on text simplification. Whilst much of previous work has focused on sentence-level simplification, text-level simplification is a promising area of research, as it represents a ‘more real use-case scenario for a simplification model’ (Alva-Manchego et al., 2020). For the German language, Leichte Sprache is a term often used in the context of simplified texts: Leichte Sprache texts are written according to clearly defined rules, however text-level aspects, including coherence, are often neglected in the guidelines, even though they are highly relevant when producing accessible texts (Bock, 2019; Maaß, 2020).

Aside from contributing to the general collection of RST annotated texts, this corpus could also be used to help carve out what text-level simplification actually constitutes, how simplified texts are structured and how this differs to their standard counterparts, and could also be used to answer questions surrounding the complexity of different types of text structures. Section 2 provides an overview on previous work that has considered questions at the intersection of discourse structure and text simplification. Section 3 provides details on the annotation process of the corpus. Section 4 outlines some statistical analysis on the corpus and Section 5 summarises the paper and provides inspiration for potential use-cases for the corpus. The corpus can be downloaded at https://github.com/fhewett/apa-rst.

2 Related work
Datasets which combine discourse structure and text simplification are relatively rare. LeiKo is a German-language corpus of newspaper articles simplified at various levels (including Leichte Sprache, Jablotschkin and Zinsmeister, 2022). A subset of 40 articles from the corpus has (manual) Penn Discourse Treebank (PDTB) annotations. Ko et al. (2023) expand their corpus of English-language texts annotated according to the Questions under Discussion (QUD) framework to include six Newsela articles and their counterparts at middle and elementary school level. In the context of text-level simplification, the task of sentence deletion has also been approached using various discourse structures. Zhang et al. (2022) also look at Newsela texts, and automatically annotate them with a ‘news genre-specific functional discourse structure’ and with sentence alignments. They train a model to predict when a sentence should be deleted and find that the discourse structure improves the accuracy. Zhong et al. (2020) also focus on the task of sentence deletion and analyse various discourse-based
features (from RST and PDTB) for this purpose and find that the position of a sentence in an RST tree as well as some specific relations play a key role. The link between discourse structure and other aspects of text-level simplification have also been considered; Siddharthan (2003) proposes a rule-based system – using cue words, for example – to help preserve coherence when restructuring texts during the simplification process. Niklaus et al. (2021, 2016) split complex sentences by using automatically parsed syntax trees. They use a pre-determined list of cue words (such as ‘although’) to determine the rhetorical relation within sentences to ensure that the split sentences are still coherent. Davoodi and Kosseim (2016) implement pairwise classification of texts of varying complexity using discourse features, using a subset of 30 articles from the PDTB which have been annotated with a complexity level (Pitler and Nenkova, 2009) and an automatically parsed subset of the Simple English Wikipedia corpus (Coster and Kaufak, 2011).

3 Corpus creation

The data used in the corpus is from the Austria Press Agency (APA), who publish four to six articles every weekday, (manually) simplified to two language levels: B1 and A2, according to the Common European Framework of Reference for Languages (CEFR). More details on the APA data can be found in Ebling et al. (2022); the version used to create the APA-RST subset contains articles up to April 2022. APA-RST covers a total of five randomly selected days from a time-frame between 2018 and 2022, with five articles each day. The corpus therefore consists of a total of 75 articles, with 25 at each level (original, B1 and A2), covering different topics such as politics, culture and sport.

3.1 RST annotations

Each article has been annotated according to the RST guidelines from Stede et al. (2017). In addition to the relations present in those guidelines, we also include two additional relations from RST-DT (Carlson and Marcu, 2001): sameunit and attribution. We also remove means from the relation set (due to its similarity to enablement) as well as un-

<table>
<thead>
<tr>
<th>Level</th>
<th>OR</th>
<th>OR</th>
<th>B1</th>
<th>A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total sent.</td>
<td>558</td>
<td>558</td>
<td>184</td>
<td>204</td>
</tr>
<tr>
<td>Total tok.</td>
<td>9567</td>
<td>9567</td>
<td>2009</td>
<td>1871</td>
</tr>
<tr>
<td>Sent./text</td>
<td>22.3</td>
<td>9.1</td>
<td>7.4</td>
<td>8.2</td>
</tr>
<tr>
<td>Tok./text</td>
<td>382.7</td>
<td>156.8</td>
<td>81.0</td>
<td>74.2</td>
</tr>
<tr>
<td>Tok./sent.</td>
<td>17.1</td>
<td>17.1</td>
<td>10.9</td>
<td>9.2</td>
</tr>
<tr>
<td>Char./tok.</td>
<td>6.2</td>
<td>6.2</td>
<td>5.8</td>
<td>5.6</td>
</tr>
<tr>
<td>EDUs/text</td>
<td>-</td>
<td>15</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 1: General information on APA-RST. Sent. stands for sentence(s), tok. for token(s), char. for characters and OR for original level. The values which are not totals represent averages.

less (due to its similarity to condition). The titles of the newspaper articles were excluded from the annotation, as well as glossary entries for complicated words, which were occasionally included in the A2 texts. Longer texts were (manually) split into separate parts for a total of 111 parts; information on these parts can be found in Table 1. These texts were segmented into EDUs and given to the annotators in pre-segmented form. Five annotators used rstWeb to annotate the texts (Zeldes, 2016). The annotators were undergraduate students of computational linguistics, who were trained for the annotation task and had regular feedback sessions during the annotation process. Approximately one third of the corpus (36 texts) has three sets of annotations. The inter-annotator agreement (IAA) was calculated using RST-Tace (Wan et al., 2019), which is based on a proposal by Iruskieta et al. (2015), and considers four different aspects: nuclearity, relations, constituents and attachment points. RST-Tace is designed for comparing two sets of annotations, so to adapt it for our three sets we simply calculated the IAA for all possible combinations, i.e. between set 1 and 2, set 1 and 3, and set 2 and 3. Overall, the average Kappa score is 0.27, and the aspect with the most disagreement between annotations was the relations. Out of all non-matching relations, elaboration and e-elaboration are the main source of disagreement, i.e. one annotator chose elaboration whilst a second annotator chose e-elaboration for the same set of EDUs. Although a certain level of subjectivity is to be expected in RST annotations, due to the relatively low agreement, all annotations were manually checked by two doctoral students. Additionally, the texts with multiple annotations

---

1The A2 level corresponds (approximately) to Leichte Sprache.
2For readers unfamiliar with RST, short descriptions of the relations mentioned in the following Sections can be found in the Appendix A.4.
have been manually harmonised and checked by one annotator from the original group and two doctoral students.

3.2 Alignment annotations

The original articles and the B1 articles have also been annotated with alignments to the simplified levels. Two annotators (an undergraduate student and a doctoral student) looked at the sentences in the B1 and A2 texts and labelled the sentence(s) in the original texts which provided the content for the simplified sentences. The inter-annotator agreement (Kappa) for aligning the sentences in the original texts to their B1 counterparts was 0.77, and 0.9 for B1 to A2. Information on the types of alignments can be found in Table 3. An example of an aligned text can be seen in Table 2; the B1 sentence [1] consists of content from two sentences in the original, this is an n:1 alignment.

4 Corpus statistics

As we can see in Table 1, the simplified texts are approximately a third of the length of the original articles, showing that simplification of newspaper articles mostly results in a shorter version. The A2 texts are slightly longer than the B1 texts, owing to more descriptions and explanations of complicated concepts.

Relation distribution. Figure 2 shows the distribution of the relations at the different complexity levels. The texts at levels B1/A2 contain more elaborations and e-elaborations than the standard texts. With regards to the multi-nuclear relations – relations which consist of two nuclei segments – the simplified texts contain more sequences and slightly more conjunctions, whereas the original texts contain almost the same amount of lists. For the original texts, the list relations are found at a higher level in the tree, as they encompass an average of 8.2 EDUs, as compared to approximately 4 in the simplified texts, as can be seen in Figure 3. The example tree in Figure 1 shows a list relation at the highest level in the tree. When it comes to causal relations, the simplified texts contain more causes, slightly more results but less reasons. According to the annotation guidelines, reason should be used to link two subjective claims, which suggests that the original texts have more subjectivity. Attribution relations occur more frequently in the standard texts. Attributing information or a quote to an external source increases the number of perspectives in a text, whereas the simplified texts have less attributions and therefore less perspectives. Sameunits do not occur at all in the simplified versions. In the original texts they are used for nested constructions; these are not present in the simplifications.

Relations and nuclearity of aligned sentences. Figure 4 shows the relations together with the nuclearity assignments for the original sentences which align with the B1 version, i.e. the original sentences which contain the content chosen for the simplification. Only those that occurred at least 5 times were included. Any bars above the line means that the corresponding relation occurs over-proportionally in the selected sentences. Elaboration N and e-elaboration N feature heavily. As the simplified texts are more concise, they mostly consist of more salient information. E-elaboration S are also selected more frequently, indicating that elaborations on specific entities or examples are useful for a simplified text. The high frequency of sequence MN also suggests that the simplified texts may have more of a linear tone; the standard deviation for the

Figure 1: The tree of the second part of an original level text (4-freitag-28-1-22-or-pt2).
In 2015, a rocket from the company SpaceX was sent into space. According to calculations by scientists at the US space agency NASA, a part of the rocket could collide with the moon in early March. The rocket was launched from the Cape Canaveral Cosmodrome in 2015 and had brought the "Deep Space Climate Observatory", an Earth observation satellite, into space.

The rocket’s fuel ran out before it could return to earth, which is why it is still in orbit. Afterwards, however, the rocket’s fuel ran out before it could return to Earth, so it’s been in orbit ever since.

According to the US space agency NASA, a part of the rocket could collide with the moon in early March. NASA announced this on Thursday. The trajectory of the "Falcon 9" rocket is currently being monitored, a NASA spokeswoman told the Deutsche Presse-Agentur on Thursday.

It will not be possible to observe the collision live. It will not be possible to observe the collision live from the "Lunar Reconnaissance Orbiter" probe, NASA said.

However, it will be investigated if there are any changes on the moon afterwards. However, it will be investigated whether changes on the moon and a possible crater created by the collision could be analysed.

The search for the crater could however take weeks or months. The search for the crater would be a major challenge and could take weeks or even months.

### Table 2: Example of an aligned text

<table>
<thead>
<tr>
<th>B1 sentences</th>
<th>Original sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] In 2015, a rocket from the company SpaceX was sent into space.</td>
<td>[1] A part of a SpaceX rocket could collide with the moon in early March, according to calculations by scientists at the US space agency NASA.</td>
</tr>
<tr>
<td>[2] The rocket’s fuel ran out before it could return to earth, which is why it is still in orbit.</td>
<td>[4] Afterwards, however, the rocket’s fuel ran out before it could return to Earth, so it’s been in orbit ever since.</td>
</tr>
<tr>
<td>[3] According to the US space agency NASA, a part of the rocket could collide with the moon in early March.</td>
<td>[1] A part of a SpaceX rocket could collide with the moon in early March, according to calculations by scientists at the US space agency NASA.</td>
</tr>
<tr>
<td>[4] NASA announced this on Thursday.</td>
<td>[2] The trajectory of the “Falcon 9” rocket is currently being monitored, a NASA spokeswoman told the Deutsche Presse-Agentur on Thursday.</td>
</tr>
<tr>
<td>[5] It will not be possible to observe the collision live.</td>
<td>[8] It will not be possible to observe the collision live from the “Lunar Reconnaissance Orbiter” probe, NASA said.</td>
</tr>
<tr>
<td>[6] However, it will be investigated if there are any changes on the moon afterwards.</td>
<td>[9] However, it will be investigated whether changes on the moon and a possible crater created by the collision could be analysed.</td>
</tr>
<tr>
<td>[7] The search for the crater could however take weeks or months.</td>
<td>[11] The search for the crater would be a major challenge and could take weeks or even months.</td>
</tr>
</tbody>
</table>

Table 2: Example of an aligned text. The B1 sentences (on the left) were aligned with the original sentence (on the right) that contains the content; the relevant content is highlighted in bold. The full original text can be found in Appendix A.1.

### Figure 2: Relation distribution at the three different complexity levels

For readability purposes, only the top 17 relations are shown (out of a total of 30).

<table>
<thead>
<tr>
<th>Level</th>
<th>1:1</th>
<th>1:n:1</th>
<th>1:n</th>
<th>1:0</th>
<th>0:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR:B1</td>
<td>85</td>
<td>5</td>
<td>33</td>
<td>430</td>
<td>30</td>
</tr>
<tr>
<td>B1:A2</td>
<td>123</td>
<td>1</td>
<td>33</td>
<td>26</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3: The types of alignments that were annotated, where n is more than one.

The depth of EDUs is in fact lower in the simplified texts (1.18 at A2 level and 1.33 at B1 level, compared to 1.4 in the original texts), indicating that the RST trees for the simplified texts are slightly more shallow. The mean depth of the trees of the original texts is 4.7 EDUs, compared to 4.1 and 3.9 for B1 and A2, respectively.

### 5 Conclusion and outlook

We have introduced a new German-language corpus of 75 parallel texts at three different complexity levels. The texts have been annotated according to the RST framework and have also been aligned at sentence level. We have shown how the relation distribution differs across the complexity levels, as well as how the relations differ in terms of what level they are used in the tree. We have also looked at the sentence alignments together with the RST annotations and shown the specific relations and nuclearity assignments of the content that is selected for a simplification. We pro-

---

3For example, the depth of the EDUs 5 and 6 in Figure 1 are 5 and 4, respectively.
Figure 3: Average (mean) amount of EDUs that the relations encompass. Only relations that occur more than once in the A2 texts are included.

Figure 4: Relations and nuclearity of aligned sentences, out of the total amount of relations. Any relation ratio above 30% (the line on the graph) is above average. Relations and nuclearity assignments occurring less than a total of 5 times are excluded. N stands for nucleus, S for satellite, MN for nucleus in a multi-nuclear relation.

we would like to thank the reviewers for their very helpful comments. This research was funded by a grant from the German Ministry of Education and Research (BMBF).

References


A.3 Original example text, B1
(4-freitag-28-1-22-b1)

A.4 Descriptions of RST relations

<table>
<thead>
<tr>
<th>Relation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>elaboration</td>
<td>‘S provides details or more information on the state of affairs described in N’</td>
</tr>
<tr>
<td>e-elaboration</td>
<td>‘S provides details or more information on a single entity mentioned in N’</td>
</tr>
<tr>
<td>sequence</td>
<td>‘the nuclei describe states of affairs that occur in a particular temporal order’</td>
</tr>
<tr>
<td>conjunction</td>
<td>‘the nuclei provide information that can be recognized as related, enumerating [...] and they are linked by coordinating conjunctions’</td>
</tr>
<tr>
<td>list</td>
<td>‘the nuclei provide information that can be recognized as related’</td>
</tr>
<tr>
<td>cause</td>
<td>‘the state/event in N is being caused by the state/event in S’</td>
</tr>
<tr>
<td>result</td>
<td>‘the state/event in S is being caused by the state/event in N’</td>
</tr>
<tr>
<td>reason</td>
<td>S and N are ‘subjective statements/theses/claims’ and ‘understanding S makes it easier for [the reader] to accept N’</td>
</tr>
<tr>
<td>attribution</td>
<td>the attribution predicate is the S, the attributed material the N</td>
</tr>
<tr>
<td>sameunit</td>
<td>used for linking two discontinuous text fragments that are really a single EDU, but which are broken up by an embedded unit</td>
</tr>
</tbody>
</table>

Table 4: These descriptions are taken from the Annotation Guidelines from Stede et al. (2017); more detailed information can be found there. The descriptions for sameunit and attribution are adapted from the RST-DT guidelines (Carlson and Marcu, 2001).
<table>
<thead>
<tr>
<th>Author Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ananiadou, Sophia</td>
<td>54</td>
</tr>
<tr>
<td>Anuranjana, Kaveri</td>
<td>99</td>
</tr>
<tr>
<td>Batista-navarro, Riza</td>
<td>54</td>
</tr>
<tr>
<td>Bleiweiss, Avi</td>
<td>145</td>
</tr>
<tr>
<td>Bürkle, Teresa</td>
<td>1</td>
</tr>
<tr>
<td>Carenini, Giuseppe</td>
<td>70</td>
</tr>
<tr>
<td>Dang, Hy</td>
<td>163</td>
</tr>
<tr>
<td>De Clercq, Orphee</td>
<td>48</td>
</tr>
<tr>
<td>De Langhe, Loic</td>
<td>48</td>
</tr>
<tr>
<td>De Montigny, Simon</td>
<td>82</td>
</tr>
<tr>
<td>Demberg, Vera</td>
<td>99</td>
</tr>
<tr>
<td>Finco, Matteo</td>
<td>1</td>
</tr>
<tr>
<td>Friedrich, Annemarie</td>
<td>1</td>
</tr>
<tr>
<td>Fu, Yuzhuo</td>
<td>106</td>
</tr>
<tr>
<td>Gao, Jingsheng</td>
<td>106</td>
</tr>
<tr>
<td>Grünwald, Stefan</td>
<td>1</td>
</tr>
<tr>
<td>Hardmeier, Christian</td>
<td>126</td>
</tr>
<tr>
<td>Henning, Sophie</td>
<td>1</td>
</tr>
<tr>
<td>Herold, Christian</td>
<td>112</td>
</tr>
<tr>
<td>Hewett, Freya</td>
<td>173</td>
</tr>
<tr>
<td>Hildebrand, Felix</td>
<td>1</td>
</tr>
<tr>
<td>Hoste, Veronique</td>
<td>48</td>
</tr>
<tr>
<td>Indrikova, Maira</td>
<td>1</td>
</tr>
<tr>
<td>Jacobs, Cassandra L.</td>
<td>156</td>
</tr>
<tr>
<td>Ji, Heng</td>
<td>24</td>
</tr>
<tr>
<td>Jiang, Meng</td>
<td>163</td>
</tr>
<tr>
<td>Knaebel, René</td>
<td>61</td>
</tr>
<tr>
<td>Lóperez Cortez, S. Magalí</td>
<td>156</td>
</tr>
<tr>
<td>Lai, Tuan</td>
<td>24</td>
</tr>
<tr>
<td>Lee, Bruce W.</td>
<td>16</td>
</tr>
<tr>
<td>Lee, Jason</td>
<td>16</td>
</tr>
<tr>
<td>Liu, Boyang</td>
<td>54</td>
</tr>
<tr>
<td>Liu, Hao</td>
<td>106</td>
</tr>
<tr>
<td>Liu, Ting</td>
<td>106</td>
</tr>
<tr>
<td>Liu, Xiao</td>
<td>37</td>
</tr>
<tr>
<td>Ney, Hermann</td>
<td>112</td>
</tr>
<tr>
<td>Ng, Victoria</td>
<td>82</td>
</tr>
<tr>
<td>Nguyen, Bang</td>
<td>163</td>
</tr>
<tr>
<td>Niu, Jingcheng</td>
<td>82</td>
</tr>
<tr>
<td>Penn, Gerald</td>
<td>82</td>
</tr>
<tr>
<td>Rees, Erin</td>
<td>82</td>
</tr>
<tr>
<td>Ruby, Ahmed</td>
<td>126</td>
</tr>
<tr>
<td>Scheffler, Tatjana</td>
<td>89</td>
</tr>
<tr>
<td>Schlegel, Viktor</td>
<td>54</td>
</tr>
<tr>
<td>Schrader, Timo</td>
<td>1</td>
</tr>
<tr>
<td>Seemann, Hannah</td>
<td>89</td>
</tr>
<tr>
<td>Shahmohammadi, Sara</td>
<td>89</td>
</tr>
<tr>
<td>Stede, Manfred</td>
<td>89</td>
</tr>
<tr>
<td>Stymne, Sara</td>
<td>126</td>
</tr>
<tr>
<td>Tan, Sherry</td>
<td>1</td>
</tr>
<tr>
<td>Varghese, Nobel</td>
<td>99</td>
</tr>
<tr>
<td>Xiang, Suncheng</td>
<td>106</td>
</tr>
<tr>
<td>Xiao, Wen</td>
<td>70</td>
</tr>
<tr>
<td>Yang, Bongseok</td>
<td>16</td>
</tr>
<tr>
<td>Yung, Frances</td>
<td>99</td>
</tr>
<tr>
<td>Zhang, Haopeng</td>
<td>37</td>
</tr>
<tr>
<td>Zhang, Jiawei</td>
<td>37</td>
</tr>
<tr>
<td>Ziems, Noah</td>
<td>163</td>
</tr>
</tbody>
</table>